I. MUNTEAN

SINTEZA AUTOMATELOR FINITE

Lucrarea prezintă o metodă unitară și riguroasă de sinteză a automatelor finite, aplicabilă atît automatelor complete cît și celor incomplet definite. Sînt prezentate de asemenea elementele de teoria mulțimilor, laticelor, algebrelor booleene, spațiilor vectoriale liniare și funcțiilor de variabile polivalente, utilizate în problema sintezei.

Lucrarea tratează toate etapele sintezei automatelor finite și prezintă pentru fiecare etapă algoritmi de rezolvare, în cea mai mare parte originali, programabili pe calculator.

Metodele și algoritmii prezentați depășesc sfera aplicațiilor tehnice, găsindu-și utilitate și în celelalte domenii ca biologia, lingvistica, clasificarea și codificarea automată, programarea și proiectarea calculatoarelor, traducerea automată, investigarea activității nervoase umane etc.

Lucrarea se adresează unui cerc larg de cititori : ingineri specialiști în teoria sistemelor, automatică, electronică, telecomunicații, informatică, matematicienilor, studenților și tuturor celor care se ocupă cu studiul unor sisteme cu număr finit de stări.

PREFAȚĂ

Automatele finite își găsesc aplicație într-un număr mare de domenii ale științei și tehnicii, de la automatizările industriale și circuitele secvențiale la programarea și proiectarea calculatoarelor electronice, traducerea automată și investigarea activității nervoase umane. În literatura de specialitate se tratează de regulă problema sintezei pentru automatele complet definite. În această lucrare se prezintă o metodă algebrică de sinteză, concepută pentru automatele incomplet definite, dar care se aplică la fel de bine pentru automatele complete.

Lucrarea încearcă o tratare completă a problemei sintezei, prezentînd, alături de noțiunile de bază privind teoria automatelor finite, descrierea detaliată și demonstrarea algoritmilor care compun etapele sintezei. Se prezintă de asemenea noțiunile matematice de algebra booleană și alte structuri algebrice, cu o dezvoltare mai mare a capitolelor privind ecuațiile booleene, ecuațiile de variabile polivalente, codificarea și simplificarea funcțiilor booleene și funcțiilor de variabile polivalente.

Prin problematica pe care o tratează și prin unele rezultate noi pe care le cuprinde, lucrarea se adresează unui cerc larg de cititori : ingineri specialiști în automatică, electronică și telecomunicații, matematicieni, informaticieni, studenți în specialitățile menționate și altor specialiști care se preocupă de aplicarea teoriei automatelor în diferite domenii ale științei și tehnicii.

AUTORUL

CUPRINS

1. **Noţiuni de bază** 9
1.1. Generalităţi 9
1.2. Elemente de teoria mulţimilor 9
1.3. Latice 12
1.4. Algebre booleene 15
1.5. Grupuri. Corpuri. 17
1.6. Spaţii liniare 19

2. **Funcţii booleene. Funcţii de variabile polivalente** 20
2.1. Funcţii booleene elementare 20
2.2. Forme normale ale funcţiilor booleene 21
2.3. Funcţii booleene inverse 23
2.4. Funcţii booleene duale 25
2.5. Funcţii booleene incomplete 26
2.6. Funcţii de variabile polivalente 28

3. **Ecuaţii booleene. Ecuaţii de variabile polivalente** 35
3.1. Egalităţi booleene 35
3.2. Ecuaţii booleene de o variabilă 35
3.3. Metoda substituţiei 36
3.4. Metoda lui Löwenheim 40
3.5. Ecuaţii de variabile polivalente 41

4. **Coduri asociate funcţiilor booleene şi funcţiilor de variabile polivalente** 45
4.1. Generalităţi 45
4.2. Coduri bivalente 46
4.3. Coduri k-valente 56
4.4. Codificarea funcţiilor booleene 64
4.5. Codificarea funcţiilor de variabile polivalente 68

5. **Reprezentările funcţiilor booleene** 73
5.1. Generalităţi 73
5.2. Reprezentarea pe hipercub 74
5.3. Reprezentarea prin scheme cu contacte de relee 76
5.4. Reprezentarea prin scheme de operatori 78

6. **Simplificarea funcţiilor booleene** 82
6.1. Introducere. Definiţii 82
6.2. Metode de simplificare bazate pe compunerea vecinilor 84
6.3. Metoda hiperfeţelor maximale 88
6.4. Metoda consensurilor 91
6.5. Metoda celui mai mare divizor comun 103

7. **Simplificarea funcţiilor de variabile polivalente** 106
7.1. Generalităţi 106
7.2. Metoda Quine- McCluskey generalizată 106
7.3. Metoda hiperfeţelor maximale 110
7.4. Metoda consensurilor generalizată 119
7.5. Metoda celui mai mare divizor comun 124

8. **Teoria abstractă a automatelor** 130
8.1. Generalităţi 130
8.2. Automate Mealy 131
8.3. Reprezentările automatelor 133
8.4. Omomorfismul şi echivalenţa automatelor 135
8.5. Automate Moore 140
8.6. Automate incomplete 144

9. **Principiul metodei lui Huffman** 148
9.1. Punerea problemei 148
9.2. Asignarea stărilor stabile 148
9.3. Stabilirea matricei de faze primitive 150
9.4 Stabilirea matricei de faze reduse 151
9.5. Afectarea variabilelor intermediare şi stabilirea matricei de excitaţii secundare 153
9.6. Condiţia tehnică de bună funcţionare. Eliminarea hazardului din circuitele de comutaţie 156
9.7. Stabilirea matricei de ieşire 158

10. **Minimizarea numărului de stări ale automatelor finite** 159
10.1. Introducere 159
10.2. Relaţia de compatibilitate a stărilor 160
10.3. Algoritmul de clasificare a stărilor 169
10.4. Definirea funcţiilor de tranziţie şi de ieşire ale automatului redus 176

11. **Asignarea variabilelor intermediare** 179
11.1. Definiţii 179
11.2. Spaţiul vectorial asociat mulţimii stărilor automatului. Variabile intermediare asociate 181
11.3. Asignarea elementelor intermediare cu respectarea condiţiei tehnice de bună funcţionare 185
11.4. Funcţiile de lucru ale elementelor intermediare 192
11.5. Eliminarea hazardului din circuitele de comutaţie 193

12. **Sinteza multipolilor fără repetiţie** 200
12.1. Generalităţi 200
12.2. Funcţia de lucru a unui multipol 201
12.3. Un algoritm de sinteză a dipolilor fără repetiţie 205
12.4. Codurile grupale generate de funcţiile booleene 208
12.5. Proprietăţi ale codurilor grupale generate de funcţiile care se realizează prin scheme fără repetiţie 210
12.6. Un criteriu de recunoaştere a funcţiilor care se realizează prin scheme fără repetiţie 212

13. **Sinteza multipolilor cu repetiție** 222
13.1. Proprietăți ale codurilor generate de funcțiile booleene care se realizează prin scheme cu repetiție 222
13.2. Alte proprietăți ale codurilor grupale asociate funcțiilor booleene 232
13.3. Funcții extinse ale funcțiilor booleene 238
13.4. Un algoritm de sinteză pentru funcțiile care se realizează prin scheme cu repetiție 247

14. **Algoritmul general de sinteză a automatelor finite** 265
14.1. Etapele sintezei automatelor finite 265
14.2. Aplicarea algoritmului general de sinteză a automatelor finite 269

Bibliografie 285

CAPITOLUL 1

NOȚIUNI DE BAZĂ

1.1. Generalități

Algebra booleană oferă posibilitatea de a descrie principalele funcțiuni ale automatelor finite prin funcții booleene. Datorită acestei proprietăți studiul automatelor finite devine mai sistematic și datorită formalizării matematice analiza și sinteza automatelor se pot face cu mijloace electronice de calcul.

În acest capitol se dau noțiunile privind mulțimile, laticele și algebrele booleene, strict necesare pentru înțelegerea capitolelor următoare. Extragerea acestor noțiuni și cuprinderea lor într-un capitol separat a urmărit evitarea încărcării capitolelor de teoria automatelor cu elemente auxiliare și simplificarea expunerii.

1.2. Elemente de teoria mulțimilor

Noțiunea de mulțime se presupune cunoscută. Dacă $\mathbf{M}$ este o mulțime și m unul din elementele sale, se notează $m \in \mathbf{M}$ și se citește „m aparține lui $\mathbf{M}$". Dacă m nu aparține lui $\mathbf{M}$, se scrie $m \notin \mathbf{M}$.

Dacă $\mathbf{M}'$ este o mulțime cu proprietatea că $m \in \mathbf{M}'$ implică $m \in \mathbf{M}$, vom spune că $\mathbf{M}'$ este *inclusă* în $\mathbf{M}$ și vom nota acest lucru prin $\mathbf{M}' \subset \mathbf{M}$. Mulțimea $\mathbf{M}'$ se va numi *submulțime* sau *parte* a lui $\mathbf{M}$.

Două mulțimi $\mathbf{M}_1$ și $\mathbf{M}_2$ sînt *egale*, dacă $\mathbf{M}_1 \subset \mathbf{M}_2$ și $\mathbf{M}_2 \subset \mathbf{M}_1$.

Dacă mulțimea $\mathbf{M}$ se dă prin elementele sale $m_1, m_2, \ldots, m_n$, vom scrie acest lucru astfel : $\mathbf{M} = \{m_1, m_2, \ldots, m_n\}$ sau $\mathbf{M} = \{m_i \mid i \in \mathbf{I}_n\}$, unde $\mathbf{I}_n = \{1, 2, \ldots, n\}$.

Mulțimea vidă, notată de obicei prin semnul $\emptyset$, este mulțimea care nu are nici un element.

Mulțimea tuturor părților lui $\mathbf{M}$ va fi notată de obicei prin $P(\mathbf{M})$. Mulțimea vidă și mulțimea $\mathbf{M}$ însăși sînt incluse în $P(\mathbf{M})$.

Principalele *operații cu mulțimile* sînt următoarele :

Reuniunea a două mulțimi $\mathbf{M}_1$ și $\mathbf{M}_2$ este mulțimea $\mathbf{M}_1 \cup \mathbf{M}_2 = \{m \mid m \in \mathbf{M}_1 \text{ sau } m \in \mathbf{M}_2\}$.

Intersecția mulțimilor $\mathbf{M}_1$ și $\mathbf{M}_2$ este mulțimea $\mathbf{M}_1 \cap \mathbf{M}_2 = \{m \mid m \in \mathbf{M}_1 \text{ și } m \in \mathbf{M}_2\}$.

Reuniunea și *intersecția* mulțimilor $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ sînt respectiv mulțimile :

$$\bigcup_{i \in \mathbf{I}_n} \mathbf{M}_i = \mathbf{M}_1 \cup \mathbf{M}_2 \cup \ldots \cup \mathbf{M}_n, \qquad \bigcap_{i \in \mathbf{I}_n} \mathbf{M}_i = \mathbf{M}_1 \cap \mathbf{M}_2 \cap \ldots \cap \mathbf{M}_n.$$

Diferența mulțimilor $\mathbf{M}_1$ și $\mathbf{M}_2$ este mulțimea $\mathbf{M}_1 \setminus \mathbf{M}_2 = \{m \mid m \in \mathbf{M}_1 \text{ și } m \notin \mathbf{M}_2\}$.

Diferența simetrică a mulțimilor $\mathbf{M}_1$ și $\mathbf{M}_2$ este mulțimea $\mathbf{M}_1 \oplus \mathbf{M}_2 = (\mathbf{M}_1 \setminus \mathbf{M}_2) \cup (\mathbf{M}_2 \setminus \mathbf{M}_1)$.

Dacă $\mathbf{M}_1 \subset \mathbf{M}_2$, atunci mulțimea $\mathbf{M}_1^c = \mathbf{M}_2 \setminus \mathbf{M}_1$ se numește *complementara* mulțimii $\mathbf{M}_1$ față de $\mathbf{M}_2$.

Produsul cartezian al mulțimilor $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ este mulțimea

$$\mathbf{M}_1 \times \mathbf{M}_2 \times \ldots \times \mathbf{M}_n = \{(m_1, m_2, \ldots, m_n) \mid m_i \in \mathbf{M}_i,\ i \in \mathbf{I}_n\}.$$

Definiția 1. *O relație binară* $\mathbf{R}$ definită pe mulțimile $\mathbf{M}_1$ și $\mathbf{M}_2$ este *o parte a produsului cartezian* $\mathbf{M}_1 \times \mathbf{M}_2$, $\mathbf{R} \subset \mathbf{M}_1 \times \mathbf{M}_2$. În cazul cînd $\mathbf{M}_1 = \mathbf{M}_2 = \mathbf{M}$ relația $\mathbf{R}$ este o relație binară pe mulțimea $\mathbf{M}$.

În general se definește o relație n-ară $\mathbf{R}$ pe mulțimile $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ prin incluziunea $\mathbf{R} \subset \mathbf{M}_1 \times \mathbf{M}_2 \times \ldots \times \mathbf{M}_n$.

Pentru ușurința exprimării, cînd nu se ivesc confuzii, o relație binară se numește simplu relație.

Dacă $x \in \mathbf{M}_1$, $y \in \mathbf{M}_2$, $\mathbf{R} \subset \mathbf{M}_1 \times \mathbf{M}_2$ și dacă $(x, y) \in \mathbf{R}$, atunci acest lucru se mai scrie și astfel : $x\,\mathbf{R}\,y$.

Un caz particular de relație binară este o aplicație de la o mulțime $\mathbf{M}_1$ la o mulțime $\mathbf{M}_2$.

Definiția 2. *O aplicație* f de la mulțimea $\mathbf{M}_1$ la mulțimea $\mathbf{M}_2$, notată $f : \mathbf{M}_1 \to \mathbf{M}_2$, este *o relație* $f \subset \mathbf{M}_1 \times \mathbf{M}_2$ cu proprietatea :

dacă $(m_1, m_2') \in f$, $(m_1, m_2'') \in f$, atunci $m_2' = m_2''$.

În general în loc de $(m_1, m_2) \in f$ se scrie $m_2 = f(m_1)$.

Dacă din $f(m_1') = f(m_1'')$ rezultă $m_1' = m_1''$, atunci se spune că f este o *aplicație injectivă* sau *injecție* sau corespondență biunivocă.

Dacă $(\forall)\ m_2 \in \mathbf{M}_2$, există $m_1 \in \mathbf{M}_1$ cu $f(m_1) = m_2$, f este *aplicație surjectivă* sau *surjecție*.

Aplicația f este *bijectivă* sau *bijecție*, dacă este injectivă și surjectivă.

Definiția 3. O relație binară, notată prin semnul $\Leftrightarrow$, definită pe o mulțime $\mathbf{M}$, se numește *relație de echivalență* dacă satisface proprietățile :

E_1. $\quad x \Leftrightarrow x$ (reflexivitate) ;

E_2. $\quad (x \Leftrightarrow y \text{ și } y \Leftrightarrow x) \Rightarrow x = y$ (simetrie) ;

E_3. $\quad (x \Leftrightarrow y \text{ și } y \Leftrightarrow z) \Rightarrow x \Leftrightarrow z$ (tranzitivitate).

Relația de echivalență joacă un rol important în reducerea numărului de stări ale automatelor finite complet definite. Acest rol rezultă din proprietatea relației de echivalență de a determina o partiție în mulțimea $\mathbf{M}$, pe care este definită, adică o împărțire a mulțimii $\mathbf{M}$ în clase nevide, disjuncte și care epuizează mulțimea $\mathbf{M}$. Într-adevăr, fie $\mathbf{M}$ o mulțime și $\Leftrightarrow$ o relație de echivalență pe mulțimea $\mathbf{M}$. Se poate demonstra

Teoremă. *Relației de echivalență* $\Leftrightarrow$ *i se atașează biunivoc o partiție a mulțimii* $\mathbf{M}$.

Demonstrație. Fie $x \in \mathbf{M}$ și mulțimea $\mathbf{M}_x = \{m \mid m \in \mathbf{M} \text{ și } m \Leftrightarrow x\}$. Mulțimea $\mathbf{M}_x$ este nevidă. Într-adevăr, $\mathbf{M}_x$ conține cel puțin elementul x, întrucît relația $\Leftrightarrow$ este reflexivă. Rezultă deci că $\bigcup_{x \in \mathbf{M}} \mathbf{M}_x = \mathbf{M}$, adică mulțimile $\mathbf{M}_x$ $x \in \mathbf{M}$, acoperă mulțimea $\mathbf{M}$. Să arătăm că acestea sînt și disjuncte, adică $\mathbf{M}_x \cap \mathbf{M}_y = \emptyset$ dacă $x \not\Leftrightarrow y$.

Presupunem contrariul. Atunci $\mathbf{M}_x \cap \mathbf{M}_y \neq \emptyset$. Fie $z \in \mathbf{M}_x \cap \mathbf{M}_y$. Dar atunci $z \in \mathbf{M}_x$ atrage $z \Leftrightarrow x$ și $z \in \mathbf{M}_y$ atrage $z \Leftrightarrow y$. Ținînd seama de tranzitivitatea relației de echivalență, ar rezulta $x \Leftrightarrow y$ contrar presupunerii, ceea ce demonstrează teorema.

Consecință. *Dacă* $x \Leftrightarrow y$, *atunci* $\mathbf{M}_x = \mathbf{M}_y$.

Definiția 4. O mulțime **P** se numește *parțial ordonată* dacă între elementele lui **P** este definită o *relație binară*, notată $\leqslant$, numită *relație de ordine parțială*, care satisface sistemul de axiome :

P_1. $x \leqslant x$ (*reflexivitate*);

P_2. $(x \leqslant y \text{ și } y \leqslant x) \Rightarrow x = y$ (*antisimetrie*);

P_3. $(x \leqslant y \text{ și } y \leqslant z) \Rightarrow x \leqslant z$ (*tranzitivitate*),

x, y, z fiind elemente din **P**, iar semnul $\Rightarrow$ se citește „implică".

O *relație de ordine parțială* între două elemente x și y este *strictă* și se notează $x < y$, dacă odată cu $x \leqslant y$ este adevărată și relația $x \neq y$.

1.3. Latice

Definirea laticelor, în mod uzual, se face pe două căi, cu ajutorul relației de ordine $\leqslant$ și cu ajutorul a două operații $\cup$ și $\cap$, cu anumite proprietăți, numite *reuniune* $\cup$ și *intersecție* $\cap$.

1.3.1. *Definirea laticelor cu ajutorul relației* $\leqslant$

Definiția 1. *O latice* este o mulțime **L** în care s-a definit o relație binară $\leqslant$ și elementele căreia satisfac următoarele proprietăți :

L_1. $x \leqslant x$;

L_2. $(x \leqslant y \text{ și } y \leqslant x) \Rightarrow (x = y)$;

L_3. $(x \leqslant y \text{ și } y \leqslant z) \Rightarrow (x \leqslant z)$;

L_4. $((\forall)\, x, y)\,((\exists)\, s)\,\{x \leqslant s \text{ și } y \leqslant s \text{ și } (x \leqslant z \text{ și } y \leqslant z) \Rightarrow s \leqslant z\}$;

L_5. $((\forall)\, x, y)\,((\exists)\, p)\,\{p \leqslant x \text{ și } p \leqslant y \text{ și } (z \leqslant x \text{ și } z \leqslant y) \Rightarrow z \leqslant p\}$.

După cum se vede, axiomele L_1, L_2, L_3 sînt identice cu P_1, respectiv P_2 și P_3. Pe de altă parte, axioma L_4 introduce noțiunea de cel mai mic majorant comun, iar axioma L_5 noțiunea de cel mai mare minorant comun a două elemente oarecare din **L**.

Cu alte cuvinte, o *latice* **L** este o mulțime parțial ordonată în care există întotdeauna cel mai mic majorant comun s și cel mai mare minorant comun p a două elemente arbitrare x și y din **L**.

1.3.2. *Definirea laticelor cu ajutorul operațiilor* $\cup$ *și* $\cap$

Definiția 2. O *latice* este o mulțime **L** în care s-au definit două operații $\cup$ și $\cap$, satisfăcînd următoarele proprietăți;

L_6. $(x \in \mathbf{L} \text{ și } y \in \mathbf{L}) \Rightarrow x \cup y \in \mathbf{L}$;

L_7. $(x \in \mathbf{L} \text{ și } y \in \mathbf{L}) \Rightarrow x \cap y \in \mathbf{L}$;

L_8. $x \cup (y \cup z) = (x \cup x) \cup z$

L_9. $x \cap (y \cap z) = (x \cap y) \cap z$ } (*asociativitate*);

L_{10}. $x \cup y = y \cup x$

L_{11}. $x \cap y = y \cap x$ } (*comutativitate*);

L_{12}. $x \cap (x \cup y) = x$

L_{13}. $x \cup (x \cap y) = x$ } (*absorbție*).

Axiomele L_6 și L_7 arată că mulțimea **L** este închisă față de operațiile $\cup$ și $\cap$, iar restul de axiome $L_8 - L_{13}$ definesc proprietățile de *asociativitate*, *comutativitate*, *absorbție* ale operațiilor $\cup$, $\cap$.

Se poate demonstra că o latice satisface în plus următoarele proprietăți :

L_{14}. $x \in \mathbf{L} \Rightarrow x \cup x \in \mathbf{L}$

L_{15}. $x \in \mathbf{L} \Rightarrow x \cap x \in \mathbf{L}$ } (idempotență).

Legătura dintre cele două definiții ale laticelor se obține notînd

$$s = x \cup y, \quad p = x \cap y.$$

1.3.3. *Latice distributive*

Definiția 3. O *latice* se numește *distributivă*, dacă verifică, în afară de axiomele $L_6 - L_{13}$, și una dintre axiomele echivalente :

D_1. $x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$;

D_2. $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$.

1.3.4. *Latice distributive cu prim și ultim element* (0 *și respectiv* 1)

Un element 0 al laticei distributive **L** este un *prim element* dacă are proprietățile :

$$x \cup 0 = x, \quad x \cap 0 = 0.$$

Un element 1 al laticei distributive **L** este un *ultim element* dacă are proprietățile :

$$x \cup 1 = 1, \quad 1 \cup x = 1, \quad x \cap 1 = x, \quad 1 \cap x = x.$$

Exemple de latice. Mulțimea numerelor naturale $\mathbf{N} = 1, 2, 3, \ldots, n, \ldots$, în care se notează prin $\leqslant$ relația de ordine naturală și în care cel mai mic majorant comun este cel mai mic multiplu comun, iar cel mai mare minorant comun este cel mai mare divizor comun, este latice.

De asemenea sînt latice mulțimea codurilor bivalente, definită în 4.2.6 și mulțimea codurilor k-valente, definită în 4.3.4.

Definiția 4. O *mulțime* **L** în care s-a definit o relație binară, ale cărei elemente satisfac proprietățile L_1, L_2, L_3 și una oricare dintre proprietățile L_4 sau L_5, *se numește semilatice.*

Exemple de semilatice sînt mulțimea funcțiilor booleene incomplete, definită în 2.5.2 și semilaticea funcțiilor de variabile polivalente incomplete, definită în 2.6.5.

1.4. Algebre booleene

1.4.1. *Definirea algebrelor booleene cu ajutorul a două operații* $+, \cdot$ *și a complementului*

O *algebră booleană* este o mulțime **B** în care s-au definit două operații binare notate $+$ și $\cdot$, numite *suma* și respectiv *produsul* a două elemente și o operație unară notată — care face ca la fiecare element x din **B** să corespundă *complementul* său, $\bar{x}$, cu următoarele proprietăți :

B_1. $(x \in \mathbf{B}$ și $y \in \mathbf{B}) \Rightarrow x + y \in \mathbf{B}$;

B_2. $(x \in \mathbf{B}$ și $y \in \mathbf{B}) \Rightarrow xy \in \mathbf{B}$;

B_3. $x + (y + z) = (x + y) + z$

B_4. $x(yz) = (xy)z$ } (*asociativitate*) ;

B_5. $x + y = y + x$

B_6. $xy = yx$ } (*comutativitate*) ;

B_7. $x + xy = x$

B_8. $x(x + y) = x$ } (*absorbție*) ;

B_9. $x + yz = (x + y)(x + z)$

B_{10}. $x(y + z) = xy + xz$ } (*distributivitate*) ;

B_{11}. $x + 0 = x$ (existența *elementului de efect nul* la adunare) ;

B_{12}. $x1 = x$ (existența *elementului de efect nul* la înmulțire) ;

B_{13}. $(\forall)\, x$, $(\exists)\, \bar{x}$, $x + \bar{x} = 1$ și $x\bar{x} = 0$ (*existența complementului*).

Se vede că operațiile $+$ și $\cdot$ au proprietățile operațiilor $\cup$ și $\cap$ din laticele distributive cu prim și ultim element, de unde rezultă că o *algebră booleană este o latice distributivă cu prim și ultim element* care are în plus proprietățile introduse de existența operației de *complementare* (de *negare*).

1.4.2. *Proprietăți ale algebrelor booleene*

Din axiomele $B_1 - B_{13}$ rezultă următoarele proprietăți ale algebrelor booleene, pe care le vom da fără demonstrație :

B_{14}. $x \cdot x = x$
B_{15}. $x + x = x$ } (*idempotență*) ;

B_{16}. $\overline{x + y} = \bar{x}\bar{y}$
B_{17}. $\overline{xy} = \bar{x} + \bar{y}$ } (*legile lui de Morgan*) ;

B_{18}. $\bar{\bar{x}} = x$ (*involuția*) ;

B_{19}. $\bar{0} = 1, \quad \bar{1} = 0$;

B_{20}. $x0 = 0x = 0, \quad x + 1 = 1 + x = 1$;

B_{21}. $(xy = x) \Leftrightarrow (x + y = y) \Leftrightarrow (\bar{x} + y = 1)$ (*implicația*) ;

B_{22} : Extensia *comutativității* și *asociativității* la un număr arbitrar, dar finit, de termeni permite efectuarea operațiilor $x_1 + x_2 + \ldots + x_n$ și $x_1 x_2 \ldots x_n$, indiferent de ordinea termenilor sau a factorilor.

B_{23}. *Principiul dualității*. Dacă se schimbă operațiile $+$ și $\cdot$ între ele, schimbînd în același timp pe 0 în 1 și pe 1 în 0, sistemul de axiome al algebrelor booleene rămîne, în afară de unele permutări, neschimbat. Rezultă că orice demonstrație rămîne valabilă dacă se fac aceste schimbări astfel încît dintr-o *teoremă* dată se obține o nouă *teoremă*, numită *duala primei teoreme*. În aceasta constă *principiul dualității* utilizat frecvent la demonstrarea unor teoreme ale algebrelor booleene.

De obicei *implicația* se notează $x \Rightarrow y$ și se citește „x implică y".

Teorema 1. *Implicația definește o relație de ordine parțială în algebra booleană* **B**.

Demonstrație. Este suficient să verificăm proprietățile mulțimilor parțial ordonate : *reflexivitatea*, *antisimetria* și *tranzitivitatea*.

Din B_{21} se deduce imediat ținînd seama de B_{10} : $(xx = x) \Leftrightarrow (x \Rightarrow x)$, ceea ce verifică *reflexivitatea*. Apoi din echivalențele $(x \Rightarrow y) \Leftrightarrow (xy = x)$, $(y \Rightarrow x) \Leftrightarrow (yx = y)$ și B_6 rezultă $x = y$, adică *antisimetria*, și, în fine, din echivalențele $(x \Rightarrow y) \Leftrightarrow (xy = x)$, $(y \Rightarrow z) \Leftrightarrow (yz = y)$ se obține succesiv : $x = xy = x(yz) = (xy)z = xz$, adică $x \Rightarrow z$, ceea ce verifică *tranzitivitatea* și demonstrează teorema 1.

Teorema 2. *Pentru orice* $x \in \mathbf{B}$ *există implicațiile* $0 \Rightarrow x \Rightarrow 1$.

Demonstrație. Implicația $0 \Rightarrow x$ rezultă din B_{20} și din B_{21}, iar $x \Rightarrow 1$ din B_{21} și B_{14}.

1.5. Grupuri. Corpuri

Definiția 1. O *mulțime nevidă* **G** se numește *grup* dacă între elementele sale s-a definit o operație binară, notată de obicei multiplicativ, care satisface următoarele proprietăți :

G_1 $x, y \in \mathbf{G} \Rightarrow xy \in \mathbf{G}$;

G_2. $x, y, z \in \mathbf{G} \Rightarrow (xy)z = x(yz)$ (asociativitate) ;

G_3. $(\forall) x \in \mathbf{G}, (\exists) e \in \mathbf{G} \Rightarrow ex = x$ (existența elementului unitate la stînga) ;

C_4. $(\forall) x \in \mathbf{G}, (\exists)\, x^{-1} \in \mathbf{G} \Rightarrow x^{-1}x = e$ (existența inversului la stînga).

Definiția 2. Un *grup* **G** care satisface în afară de proprietățile G_1, G_2, G_3 și G_4 proprietatea

G_5. $xy \in \mathbf{G}\ xy = yx$ (comutativitate)

se numește *grup comutativ* sau *abelian*.

Pentru grupurile abeliene se utilizează de obicei notația aditivă, astfel că operația grupală se notează prin semnul $+$, elementul invers al unui element $x \in \mathbf{G}$ se notează prin $-x$ în loc de x^{-1}, iar elementul unitate numit și element neutru se notează prin 0.

Se pot găsi foarte multe exemple de grupuri. Astfel, mulțimea numerelor întregi față de operația de adunare și respectiv mulțimea numerelor raționale, fără zero, față de operația de înmulțire, formează grupuri.

În această lucrare se tratează pe larg codurile grupale generate de funcțiile booleene, introduse în 12.4, care joacă un rol important în algoritmul general de sinteză al automatelor.

În lucrare se folosește de asemenea noțiunea de pseudogrup introdusă în 4.3.3. Un pseudogrup este un grup neasociativ.

Definiția 3. *Se numește corp* o mulțime nevidă $\mathbf{K}$ prevăzută cu două operații:

I. Adunarea, față de care $\mathbf{K}$ formează *grup abelian*, deci îndeplinește proprietățile:

A_1. $x, y \in \mathbf{K} \Rightarrow x + y \in \mathbf{K}$;

A_2. $x, y, z \in \mathbf{K} \Rightarrow x + (y + z) = (x + y) + z$;

A_3. $(\forall)\, x \in \mathbf{K},\ (\exists)\, 0 \in \mathbf{K} \Rightarrow 0 + x = x$;

A_4. $(\forall)\, x \in \mathbf{K},\ (\exists) - x \in \mathbf{K} \Rightarrow (-x) + x = 0$;

A_5. $x, y \in \mathbf{K} \Rightarrow x + y = y + x$.

II. Înmulțirea, față de care $\mathbf{K} - \{0\}$ *formează grup*, deci sînt îndeplinite proprietățile G_1, G_2, G_3 și G_4.

III. Cele două operații sînt legate prin două *legi de distributivitate* la *stînga* și la *dreapta*:

D_1. $x, y, z \in \mathbf{K} \Rightarrow x(y + z) = xy + xz$;

D_2. $x, y, z \in \mathbf{K} \Rightarrow (x + y)z = xz + yz$.

Dacă $\mathbf{K} - \{0\}$ formează grup abelian față de înmulțire, atunci $\mathbf{K}$ se numește *corp comutativ* sau *cîmp*.

Exemple de corpuri: mulțimea numerelor raționale formează corp comutativ la fel ca și mulțimea numerelor reale. Mulțimea $\{0,1\}$ formează corp comutativ în care operațiile sînt adunarea și înmulțirea, operațiile obișnuite din mulțimea numerelor reale.

1.6. Spații liniare

Definiție. O mulțime nevidă $\mathbf{R}$ se numește *spațiu liniar afin*, peste un *corp* $\mathbf{K}$, dacă este prevăzută cu două operații:

I. *Operația internă, adunarea*, față de care $\mathbf{R}$ formează *grup abelian*, deci sînt îndeplinite proprietățile $A_1 - A_5$ date în 1.5.

II. *Operația externă, înmulțirea cu scalari*, față de care sînt îndeplinite proprietățile:

R_1. $x \in \mathbf{R}$ și $a \in \mathbf{K} \Rightarrow ax \in \mathbf{R}$;

R_2. $1 \in \mathbf{K}$ și $x \in \mathbf{R} \Rightarrow 1x = x$;

R_3. $a, b \in \mathbf{K}$ și $x \in \mathbf{R} \Rightarrow a(bx) = (ab)x$ (asociativitatea mixtă);

R_4. $a, b \in \mathbf{K}$ și $x \in \mathbf{R} \Rightarrow (a+b)x = ax + bx$;

R_5. $a \in \mathbf{K}$ și $x, y \in \mathbf{R} \Rightarrow a(x + y) = ax + ay$.

Proprietățile R_4 și R_5 leagă cele două operații între ele.

Un *spațiu liniar afin* se mai numește și *spațiu vectorial*, iar elementele lui se numesc *vectori*. Un exemplu de spațiu liniar este spațiul liniar real în care corpul $\mathbf{K}$ este corpul numerelor reale.

În această lucrare un rol important joacă spațiul liniar $\mathbf{B}_2^n$ al vectorilor bivalenți cu n componente peste corpul format din mulțimea $\{0, 1\}$, definit în 4.2.4.

FUNCȚII BOOLEENE. FUNCȚII DE VARIABILE POLIVALENTE

2.1. Funcții booleene elementare

După cum se știe, există patru funcții de o variabilă, constantele 0 și 1, x și $\bar{x}$ și 16 funcții de două variabile. În general există 2^{2^n} funcții booleene de n variabile.

Deși în axiomatica algebrelor booleene apar trei operații: *negația*, *reuniunea* și *intersecția*, se poate demonstra că pentru definirea acestora sînt suficiente numai două: *negația* și *reuniunea* sau *negația* și *intersecția*, după cum rezultă din relațiile

$$xy = \overline{\bar{x} + \bar{y}}, \qquad x + y = \overline{\bar{x}\,\bar{y}}.$$

Pentru studiul automatelor finite prezintă interes deosebit următoarele cinci funcții de două variabile:

a) funcția *sau exclusiv* (*dilema*, *suma modulo* 2), notată de obicei cu simbolul $\oplus$ și definită de expresia $x \oplus y = x\bar{y} + \bar{x}y$;

b) funcția *identitate* (*echivalența*), notată cu simbolul $\odot$ și dată de expresia $x \odot y = xy + \bar{x}\bar{y}$;

c) funcția *nici*, notată cu simbolul $\top$ și definită de expresia $x \top y = \bar{x}\bar{y}$;

d) funcția *incompatibilitate*, notată cu simbolul $\perp$ și definită de expresia $x \perp y = \bar{x} + \bar{y}$;

e) funcția *implicație*, notată cu simbolul $\Rightarrow$ și definită de expresia $x \Rightarrow y = \bar{x} + y$.

Pierce și Sheffer au arătat că este posibil să se exprime toate funcțiile booleene numai cu ajutorul unei singure funcții, funcția *nici*, sau funcția *incompatibilitate*.

Pentru demonstrație este suficient să exprimăm în funcție de acestea *reuniunea*, *intersecția* și *negația*. De exemplu, pentru funcția *nici* se obține

$$\bar{x} = x \top x, \; x + y = (x \top y) \top (x \top y),$$
$$x \cdot y = (x \top x) \top (y \top y).$$

De asemenea, pentru funcția *incompatibilitate* se obține

$$\bar{x} = x \perp x, \quad x + y = (x \perp x) \perp (y \perp y),$$
$$x \cdot y = (x \perp y) \perp (x \perp y).$$

2.2. Forme normale ale funcțiilor booleene

2.2.1. *Generalități*

Fie $f(x_1, x_2, \ldots, x_n)$ o funcție booleană de n variabile booleene $x_1, x_2, \ldots, x_n$. Vom asocia variabilelor $x_1, x_2, \ldots, x_n$ variabilele booleene $a_1, a_2, \ldots, a_n$ și vom scrie $x_k^{a_k} = x_k$ pentru $a_k = 1$ și $x_k^{a_k} = \bar{x}_k$ pentru $a_k = 0$. Rezultă deci $x_k = x_k^1$, $\bar{x}_k = x_k^0$, $1^1 = 0^0 = 1$, $1^0 = 0^1 = 0$.

Fie $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ și $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ vectorii booleeni asociați mulțimilor de variabile booleene $\{x_1, x_2, \ldots, x_n\}$ și respectiv $\{a_1, a_2, \ldots, a_n\}$. Întrucît $x_1, x_2, \ldots, x_n$ pot lua valorile 0 sau 1, există 2^n vectori $\mathbf{X}$. Vom nota prin $\mathbf{B}_2^n$ mulțimea acestor vectori. În cap. 4 vom demonstra că $\mathbf{B}_2^n$ formează spațiu vectorial peste corpul $\mathbf{K} = \{0, 1\}$.

2.2.2. *Prima formă normală. Forma normală disjunctivă*

Fie produsul $P_a(\mathbf{X}) = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$. Evident $P_x(\mathbf{X}) = 1$.

Fie $\mathbf{K}$ o submulțime a lui $\mathbf{B}_2^n$ și $\overline{\mathbf{K}}$ complementara lui $\mathbf{K}$ față de $\mathbf{B}_2^n$, $\mathbf{K}$, $\overline{\mathbf{K}} \subset \mathbf{B}_2^n$, $\mathbf{K} \cup \overline{\mathbf{K}} = \mathbf{B}_2^n$, $\mathbf{K} \cap \overline{\mathbf{K}} = \emptyset$.

Vom defini o funcție booleană de n variabile $f(\mathbf{X}) = f(x_1, x_2, \ldots, x_n)$, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ astfel:

$$\mathbf{X} \in \mathbf{K} \Rightarrow f(\mathbf{X}) = 1, \qquad \mathbf{X} \in \overline{\mathbf{K}} \Rightarrow f(\mathbf{X}) = 0.$$

Unei funcții booleene i se asociază deci un vector $\mathbf{V}_f = (f(\mathbf{X}))$ cu 2^n componente egale cu 0 sau 1, fiecare componentă fiind asociată unui vector $\mathbf{X}$ dat.

Consecință. *Întrucît există 2^{2^n} vectori bivalenți cu 2^n componente, rezultă că mulțimea funcțiilor booleene de n variabile are 2^{2^n} elemente.*

Lema 1. *Funcția booleană de n variabile $f(x_1, x_2, \ldots, x_n)$ se poate scrie sub forma* $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbf{B}_2^n} f(\mathbf{a})\, P_a(\mathbf{X})$.

Demonstrație. Considerăm expresia $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbf{B}_2^n} C(\mathbf{a})\, x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n} = \sum_{\mathbf{a} \in \mathbf{B}_2^n} C(\mathbf{a})\, P_a(\mathbf{X})$, $(\mathbf{X} = \mathbf{a} \Leftrightarrow P_a(\mathbf{X}) = 1, \mathbf{X} \neq \mathbf{a} \Leftrightarrow P_a(\mathbf{X}) = 0) \Rightarrow C(\mathbf{a}) = f(\mathbf{a})$, $(\forall)\, \mathbf{a} \in \mathbf{B}_2^n$, ceea ce trebuia demonstrat.

Definiție. Forma $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbf{B}_2^n} f(\mathbf{a})\, x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ a funcției booleene $f(x_1, x_2, \ldots, x_n)$ se numește *formă normală canonică disjunctivă.*

2.2.3. *A doua formă normală. Forma normală conjunctivă*

Fie $S_a(\mathbf{X}) = \overline{P}_{\bar{a}}(\mathbf{X})$. Rezultă $S_a(\mathbf{X}) = \sum_{i \in [1, n]} x_i^{a_i}$, unde $[1, n] = \{1, 2, \ldots, n\}$.

Lema 1. *Funcția booleană de n variabile $f(x_1, x_2, \ldots, x_n)$ se poate scrie sub forma* $f(x_1, x_2, \ldots, x_n) = \prod_{\mathbf{a} \in \mathbf{B}_2^n} [f(\mathbf{a}) + x_1^{\bar{a}_1} + x_2^{\bar{a}_2} + \ldots + x_n^{\bar{a}_n}] = \prod_{\mathbf{a} \in \mathbf{B}_2^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, $S_{\bar{a}}(\mathbf{X}) = x_1^{\bar{a}_1} + \ldots + x_n^{\bar{a}_n}$.

Demonstrație. Considerăm expresia $f(x_1, x_2, \ldots, x_n) = \prod_{\mathbf{a} \in \mathbf{B}_2^n} [C(\mathbf{a}) + x_1^{a_1} + x_2^{\bar{a}_2} + \ldots + x_n^{\bar{a}_n}] = \prod_{\mathbf{a} \in \mathbf{B}_2^n} [C(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$, $(\mathbf{X} = \mathbf{a} \Leftrightarrow S_{\bar{a}}(\mathbf{X}) = 0, \mathbf{X} \neq \mathbf{a} \Leftrightarrow S_{\bar{a}}(\mathbf{X}) = 1) \Rightarrow C(\mathbf{a}) = f(\mathbf{a})$, $(\forall)\, \mathbf{a} \in \mathbf{B}_2^n$, ceea ce trebuia demonstrat.

Definiție. Forma $f(x_1, x_2, \ldots, x_n) = \prod_{\mathbf{a} \in \mathbf{B}_2^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})] = \prod_{\mathbf{a} \in \mathbf{B}_2^n} [f(\mathbf{a}) + x_1^{\bar{a}_1} + x_2^{\bar{a}_2} + \ldots + x_n^{\bar{a}_n}]$ a funcției booleene $f(x_1, x_2, \ldots, x_n)$ se numește *forma normală canonică conjunctivă.*

Exemplu. $f(x_1, x_2, x_3) = x_1(x_2 + x_3)$ are următoarele forme normale canonice:

$$x_1(x_2 + x_3) = x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3 =$$
$$= (\bar{x}_1 + x_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + x_2 + x_3).$$

2.2.4. *Forme normale necanonice*

Definiția 1. Scrierea funcțiilor booleene sub formă de sumă de produse de variabile se numește *forma normală disjunctivă* sau *prima formă normală* a funcțiilor booleene.

Definiția 2. Scrierea funcțiilor booleene sub formă de produse de sume de variabile se numește *formă normală conjunctivă* sau *a doua formă* normală a funcțiilor booleene.

În timp ce formele normale canonice sînt unice, există, în general, mai multe forme normale necanonice. De exemplu, funcția majoritară de trei variabile se poate scrie sub următoarele forme normale necanonice: $MAJ(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 = (x_1 + x_2)(x_1 + x_3)(x_2 + x_3) = x_1x_2\bar{x}_3 + x_1x_3 + x_2x_3$.

2.3. Funcții booleene inverse

Definiție. Funcția $F(x_1, x_2, \ldots, x_n)$ este *inversa* funcției $f(x_1, x_2, \ldots, x_n)$ dacă sînt satisfăcute relațiile $f(x_1, x_2, \ldots, x_n) + F(x_1, x_2, \ldots, x_n) = 1$, $f(x_1, x_2, \ldots, x_n) \cdot F(x_1, x_2, \ldots, x_n) = 0$, $(\forall)\, \mathbf{x} \in \mathbf{B}_2^n$.

Lema. *Inversa unei funcții booleene* $f(x_1, x_2, \ldots, x_n)$ *se obține aplicînd acestei funcții operatorul* $^-$, *adică* $F(x_1, x_2, \ldots, x_n) = \bar{f}(x_1, x_2, \ldots, x_n)$.

Demonstrație. Fie $f(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$, $\bar{f}(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} \bar{f}(\mathbf{a})\, S_{\bar{a}}(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} \bar{f}(\mathbf{a})\, P_a(\mathbf{X})$. Dar atunci $f(\mathbf{X}) + \bar{f}(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} f(\mathbf{a})\, P_a(\mathbf{X}) + \sum_{\mathbf{a}\in\mathbf{B}_2^n} \bar{f}(\mathbf{a})\, P_a(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} [f(\mathbf{a}) + \bar{f}(\mathbf{a})]\, P_a(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} P_a(\mathbf{X}) = 1$.

Fie apoi $f(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} f(\mathbf{a})\, P_a(\mathbf{X}) \Rightarrow \bar{f}(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + \bar{P}_a(\mathbf{X})] = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$, $f(\mathbf{X})\bar{f}(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})] \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})] = \prod_{\mathbf{a}\in\mathbf{B}_2^n} S_{\bar{a}}(\mathbf{X}) = 0$,

de unde rezultă că $\bar{f}(\mathbf{X})$ îndeplinește condițiile funcției inverse, ceea ce trebuia demonstrat.

Cealaltă formă a funcției inverse, *forma normală conjunctivă*, este

$$f(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} f(\mathbf{a})\, P_a(\mathbf{X}) \Rightarrow \bar{f}(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})].$$

Exemplu. Funcția inversă a funcției $x_1(x_2 + x_3)$ admite următoarele forme normale canonice:

$$\overline{x_1(x_2 + x_3)} = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1\bar{x}_2\bar{x}_3 = \bar{x}_1 + \bar{x}_2\bar{x}_3,$$

$$\overline{x_1(x_2 + x_3)} = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + \bar{x}_2 + x_3)(\bar{x}_1 + x_2 + \bar{x}_3) = \bar{x}_1 + \bar{x}_2\bar{x}_3.$$

Se observă că pentru această funcție este avantajos să se folosească forma normală conjunctivă, care are o scriere mai simplă decît forma normală disjunctivă. Ținînd seama că inversa unei funcții este complet determinată de funcția dată și reciproc, în practică se poate folosi cea mai simplă dintre cele două funcții inverse.

2.4. Funcții booleene duale

În aplicațiile practice ale funcțiilor booleene prezintă mare interes utilizarea funcțiilor duale, obținute prin aplicarea principiului dualității la funcțiile booleene date. Acest interes este determinat atît de ușurința unor demonstrații cît și de avantajele pe care le vor aduce în reprezentările funcțiilor booleene, după cum se va vedea în capitolele următoare.

Definiție. Se numește *funcție duală* a funcției booleene $f(x_1, x_2, \ldots, x_n)$ funcția booleeană notată $f^*(x_1, x_2, \ldots, x_n)$ sau simplu $f^*(\mathbf{X})$, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, obținută prin aplicarea principiului dualității funcției $f(\mathbf{X})$:

$$f(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} f(\mathbf{a})\, P_a(\mathbf{X}) \Rightarrow f^*(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_a(\mathbf{X})].$$

La fel rezultă și cealaltă formă a funcției duale, forma normală disjunctivă

$$f(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})] \Rightarrow f^*(\mathbf{X}) = \sum_{\mathbf{a}\in\mathbf{B}_2^n} \bar{f}(\mathbf{a})\, P_a(\mathbf{X}).$$

Lemă. *Între funcția duală și funcția inversă ale unei funcții booleene există relația* $f^*(\mathbf{X}) = \bar{f}(\bar{\mathbf{X}})$.

Demonstrație. Este suficient să verificăm această proprietate pe una din formele funcției duale și una din formele funcției inverse, întrucît restul de forme sînt echivalente cu acestea.

Într-adevăr $\bar{f}(\mathbf{X}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})] \Rightarrow \bar{f}(\bar{\mathbf{X}}) = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\bar{\mathbf{X}})] = \prod_{\mathbf{a}\in\mathbf{B}_2^n} [\bar{f}(\mathbf{a}) + S_a(\mathbf{X})] = f^*(\mathbf{X})$.

Exemplu. Duala funcției $f(x_1, x_2, x_3) = x_1(x_2 + x_3)$ are următoarele forme canonice normale:

$$f^*(x_1, x_2, x_3) = (x_1 + x_2x_3)^* = (x_1 + \bar{x}_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + x_2 + x_3) =$$

$$= \bar{x}_1x_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3.$$

2.5. Funcţii booleene incomplete

2.5.1. *Definiţie şi generalităţi*

Funcţiile booleene de care ne-am ocupat în alineatele precedente s-au bucurat de proprietatea că coeficienţii $f(\mathbf{a})$ din forma canonică normală disjunctivă şi respectiv din forma canonică normală conjunctivă sînt presupuşi cunoscuţi în totalitatea lor. În aplicaţiile practice, se întîlnesc cazuri cînd unii dintre coeficienţii formelor normale nu sînt definiţi. Din punctul de vedere al aplicaţiilor respective aceşti coeficienţi pot lua valori arbitrare 0 sau 1.

Definiţie. O funcţie booleană $f(x_1, x_2, \ldots, x_n)$ pentru care unii coeficienţi ai formelor canonice normale nu sînt definiţi se numeşte *funcţie incomplet definită* sau simplu *incompletă*.

Să considerăm ordinea parţială indusă de implicaţie şi funcţia $f_0(x_1, x_2, \ldots, x_n)$ obţinută din $f(x_1, x_2, \ldots, x_n)$ prin egalarea cu 0 a coeficienţilor nedefiniţi şi funcţia $f_1(x_1, x_2, \ldots, x_n)$ obţinută din $f(x_1, x_2, \ldots, x_n)$ prin egalarea cu 1 a coeficienţilor nedefiniţi. Atunci evident $f_0 \Rightarrow f_1$, ceea ce vom nota astfel : $f_0 \leqslant f_1$.

Orice funcţie g, astfel încît $f_0 \leqslant g \leqslant f_1$, poate fi obţinută din funcţia incompletă f prin particularizarea coeficienţilor nedefiniţi. Atunci f poate fi interpretată ca mulţimea funcţiilor g care satisfac relaţiile $f_0 \leqslant g \leqslant f_1$, unde f_0 reprezintă *marginea inferioară* a acestor funcţii iar f_1 *marginea superioară* a lor.

Lema. *Fie $f = (f_0, f_1)$ o funcţie booleană incompletă. Atunci* $f_0 \leqslant g \leqslant f_1 \Rightarrow (\bar{f}_1 \leqslant \bar{g} \leqslant \bar{f}_0,\ f_1^* \leqslant g^* \leqslant f_0^*)$.

Demonstraţie. $f_0 \leqslant g \leqslant f_1 \Rightarrow (f_0 \bar{g} = 0,\ g \bar{f}_1 = 0) \Rightarrow (\bar{f}_1 \leqslant \bar{g} \leqslant \bar{f}_0)$. Apoi $(\bar{f}_1(\mathbf{X}) = f_1^*(\mathbf{X}),\ \bar{g}(\mathbf{X}) = g^*(\mathbf{X}),\ \bar{f}_0(\mathbf{X}) = \bar{f}_0^*(\mathbf{X}),\ \bar{f}_1 \leqslant \bar{g} \leqslant \bar{f}_0) \Rightarrow f_1^* \leqslant g^* \leqslant f_0^*$, ceea ce demonstrează lema.

Vom nota $\bar{f} = (\bar{f}_1, \bar{f}_0)$ şi $f^* = (f_1^*, f_0^*)$ funcţiile booleene incomplete astfel obţinute.

Definiţia 1. Funcţia booleană incompletă $\bar{f} = (\bar{f}_1, \bar{f}_0)$ se va numi *funcţia incompletă inversă* a funcţiei incomplete $f = (f_0, f_1)$.

Ţinînd seama de lema de mai sus, rezultă că $\bar{f}$ este *mulţimea inverselor* funcţiilor din intervalul $f_0 \leqslant f_1$.

Definiţia 2. Funcţia booleană incompletă $f^* = (f_1^*, f_0^*)$ se va numi *funcţia incompletă duală* a funcţiei incomplete $f = (f_0, f_1)$, iar f^* *mulţimea dualelor* funcţiilor g, cuprinse în intervalul $f_0 \leqslant f_1$.

2.5.2. *Semilaticea funcţiilor booleene incomplete*

După cum am văzut, mulţimea funcţiilor booleene incomplete este definită de mulţimea perechilor de funcţii booleene comparabile, în raport cu relaţia de implicaţie. Fie $\mathbf{F}_B^i$ această mulţime. Orice funcţie booleană poate fi interpretată ca funcţie incompletă, întrucît din $f \leqslant f$ se deduce $f = (f, f)$.

Dacă notăm cu $\mathbf{F}_B$ mulţimea funcţiilor booleene, atunci există relaţia de incluziune $\mathbf{F}_B \subset \mathbf{F}_B^i$, respectiv $\mathbf{F}_B^i$ este o extindere a lui $\mathbf{F}_B$.

Fie $f = (f_0, f_1)$, $g = (g_0, g_1)$ două funcţii incomplete. În $\mathbf{F}_B^i$ vom introduce o relaţie de ordine parţială, după cum urmează : $f \leqslant g \Leftrightarrow (g_0 \leqslant f_0, f_1 \leqslant g_1)$ şi operaţia de adunare definită astfel :

$$f + g = (f_0 g_0, f_1 + g_1).$$

Teorema 1. *Mulţimea funcţiilor incomplete $\mathbf{F}_B^i$ poate fi organizată ca semilatice faţă de adunare cu ajutorul relaţiei de ordine parţială, indusă de implicaţie.*

Demonstraţie. Să arătăm mai întîi că relaţia $f \leqslant g$ este o relaţie de ordine parţială. Într-adevăr :

L_1. $(f_0 \leqslant f_0,\ f_1 \leqslant f_1) \Rightarrow f \leqslant f$.

L_2. 1° $f \leqslant g \Leftrightarrow (g_0 \leqslant f_0, f_1 \leqslant g_1)$.

2° $g \leqslant f \Leftrightarrow (f_0 \leqslant g_0, g_1 \leqslant f_1)$.

Din 1° și 2° se deduce $(f_0 = g_0,\ f_1 = g_1) \Rightarrow f = g$.

L_3. 1° $f \leqslant g \Leftrightarrow (g_0 \leqslant f_0,\ f_1 \leqslant g_1)$.

2° $g \leqslant h \Leftrightarrow (h_0 \leqslant g_0,\ g_1 \leqslant h_1)$.

Din 1° și 2° se deduce : $(h_0 \leqslant f_0,\ f_1 \leqslant h_1) \Rightarrow f \leqslant h$.

L_4. Fie $f, g \in \mathbf{F}_B^i$ și fie $s = (f_0 g_0, f_1 + g_1)$.

Evident : $f_0 g_0 \leqslant f_1 + g_1$. Într-adevăr,

$$\overline{f_0 g_0} + f_1 + g_1 = \bar{f}_0 + \bar{g}_0 + f_1 + g_1 = 1 \Rightarrow s \in \mathbf{F}_B^i,$$

$$f \leqslant s, \qquad g \leqslant s.$$

Fie $h = (h_0, h_1) \in \mathbf{F}_B^i$ astfel încît $f \leqslant h, g \leqslant h$. Atunci vom avea succesiv : $(h_0 \leqslant f_0,\ h_0 \leqslant g_0) \Rightarrow h_0 \leqslant f_0 g_0$, $(f_1 \leqslant h_1,\ g_1 \leqslant h_1) \Rightarrow f_1 + g_1 \leqslant h_1$, adică L_4 este îndeplinită, ceea ce demonstrează teorema.

Semilaticea funcțiilor booleene incomplete admite ca ultim element pe (0, 1) după cum se poate verifica ușor. Într-adevăr, dacă $f = (f_0, f_1)$ atunci $0 \leqslant f_0 \leqslant f_1 \leqslant 1$.

Se poate demonstra cu ușurință că axioma L_5 nu este îndeplinită de mulțimea $\mathbf{F}_B^i$. Dacă în schimb luăm în considerare numai submulțimea funcțiilor booleene $\mathbf{F}_B$ a lui $\mathbf{F}_B^i$, atunci pentru aceasta L_5 va fi îndeplinită.

Teorema 2. *Mulțimea funcțiilor booleene poate fi organizată ca latice cu ajutorul relației de ordine parțială, indusă de implicație.*

Teorema 2 este, de fapt, o afirmație mai puternică decît teorema 1, o continuare a acesteia, astfel că verificarea axiomelor L_1, L_2, L_3 s-a făcut prin demonstrarea teoremei 1. Verificarea lemelor L_4 și L_5 se face de asemenea imediat, luînd $s = f + g$ și $p = fg$.

2.6. Funcții de variabile polivalente

2.6.1. *Definiție, generalități*

Teoria clasică, bivalentă se dovedește nesatisfăcătoare pentru descrierea funcționării a automatelor, chiar atunci cînd se iau în considerare numai elemente bipoziționale. Într-adevăr, dacă de exemplu, paleta unui releu ia două poziții, una de repaus și una de lucru, la trecerea de la o poziție la alta se produce o discontinuitate care poate să

provoace deranjamente în funcționarea automatului. Pentru evitarea hazardului este util să se asocieze releului o variabilă trivalentă, două valori ale acesteia indicînd stările de lucru și de repaus, iar a treia poziție intermediară. Pe de altă parte, în multe cazuri, automatele se pot realiza mai simplu prin elemente multipoziționale. De aceea în acest capitol vom dezvolta, pe scurt, teoria funcțiilor de variabile polivalente.

Definiție. O variabilă care poate lua k valori va fi numită *variabilă k-valentă*. Dacă notăm cu $0, 1, 2, \ldots, k-1$ cele k valori ale variabilei k-valente x, atunci este, adeseori, convenabil să reprezentăm variabila x prin k variabile bivalente x^a, $a = 0, 1, 2, \ldots, k-1$, cu proprietățile $x^a = 1$ pentru $x = a$ și $x^a = 0$ pentru $x \neq a$.

Rezultă imediat următoarele proprietăți : $\sum_{a=0}^{k-1} x^a = 1$, $x^a x^b = 0$ pentru $a \neq b$ și pentru orice x.

Definiție. Numim *funcție de variabile polivalente* o funcție de n variabile $f(x_1, x_2, \ldots, x_n)$ care ia valori în L_2 și în care variabilele x_i, $i = 1, 2, \ldots, n$ sînt polivalente. Variabila x_i este k_i-valentă $i = 1, 2, \ldots, n$ și în general $k_i \neq k_j$ pentru $i \neq j$.

Funcțiile de variabile bivalente sînt cazuri particulare de funcții de variabile polivalente, astfel încît unele din proprietățile acestor funcții se vor regăsi printre proprietățile funcțiilor de variabile polivalente.

Reprezentarea variabilelor polivalente prin variabile bivalente permite elaborarea unei teorii unitare a funcțiilor de variabile polivalente și bivalente, precum și o extindere naturală a algebrelor booleene.

2.6.2. Forme normale ale funcțiilor de variabile polivalente

2.6.2.1. *Forma normală disjunctivă canonică*

Fie $f(x_1, x_2, \ldots, x_n)$ o funcție de n variabile polivalente. Vom asocia variabilelor $x_1, x_2, \ldots, x_n$, n variabile polivalente $a_1, a_2, \ldots, a_n$, unde a_i, pentru orice i, va lua

valorile 0, 1, 2, ..., k_i-1 dacă x_i este k_i-valentă și unde $x_i^{a_i} = 1$ pentru $x_i = a_i$, $x_i^{a_i} = 0$ pentru $x_i \neq a_i$. Notăm prin $\mathbf{V}_i = \{0, 1, 2, \ldots, k_i - 1\}$ mulțimea valorilor variabilei x_i, $(\forall)\ i \in \mathbf{I}_n$ și $\mathbf{V} = (v_1, v_2, \ldots, v_n)$, $v_i \in \mathbf{V}_i$, $i \in \mathbf{I}_n$.

Ca și în cazul bivalent vom considera produsul $P_a(\mathbf{X}) = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, căruia îi vom asocia o variabilă booleană $f(\mathbf{a})$.

Întrucît a_i, $i = 1, 2, \ldots, n$, pot lua k_i valori, există un număr de $N_k^n = \prod_{i=1}^{n} k_i$ vectori polivalenți $\mathbf{a}$ cărora le corespund N_k^n produse diferite de forma $P_a(\mathbf{X})$ și tot atîtea variabile booleene $f(\mathbf{a})$. Vom nota cu $\mathbf{B}_k^n$ mulțimea vectorilor n valenți.

Fie $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbf{B}_k^n} f(\mathbf{a})\, P_a(\mathbf{X})$.

Lemă. *Funcția booleană de n variabile polivalente* $f(x_1, x_2, \ldots, x_n)$ *se poate scrie sub forma* $f(\mathbf{X}) = \sum_{\mathbf{a} \in \mathbf{B}_k^n} f(\mathbf{a})\, P_a(\mathbf{X})$, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$.

Demonstrație. Considerăm expresia $f(\mathbf{X}) = \sum_{\mathbf{a} \in \mathbf{B}_k^n} C(\mathbf{a})\, P_a(\mathbf{X})$, $(\mathbf{a} = \mathbf{X} \Rightarrow P_a(\mathbf{X}) = 1,\ \mathbf{a} \neq \mathbf{X} \Rightarrow P_a(\mathbf{X}) = 0) \Rightarrow C(\mathbf{a}) = f(\mathbf{a})$.

Rezultă că o funcție booleană de variabile polivalente este complet determinată de cei N_k^n coeficienți $f(\mathbf{a})$, altfel spus, este determinată de un vector boolean de dimensiunea N_k^n.

Numărul de vectori booleeni de dimensiune N_k^n este egal cu $2^{N_k^n}$, de unde rezultă că există tot atîtea funcții booleene de variabile polivalente de tipul $(k_1, k_2, \ldots, k_n)$, unde prin $(k_1, k_2, \ldots, k_n)$ am notat vectorul k-valențelor variabilelor polivalente $x_1, x_2, \ldots, x_n$.

Definiție. Expresia $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbf{B}_k^n} f(\mathbf{a}) P_a(\mathbf{X})$ se numește *forma normală disjunctivă canonică* a funcțiilor de variabile polivalente.

2.6.2.2. **Forma normală conjunctivă canonică**

Să considerăm sumele

$$S_a(\mathbf{X}) = \sum_{i \in \mathbf{I}_n} x_i^{a_i}, \qquad S_{\bar{a}}(\mathbf{X}) = \sum_{i \in \mathbf{I}_n} x_i^{\bar{a}_i}, \qquad x_i^{\bar{a}_i} = \sum_{u \neq a_i} x_i^u,$$

care au următoarele proprietăți: $x_i^{a_i} + x_i^{\bar{a}_i} = 1$, $x_i^{a_i} x_i^{\bar{a}_i} = 0$, $S_a(\mathbf{X}) = 1$ pentru $\mathbf{X} = \mathbf{a}$ și $S_a(\mathbf{X}) = 0$ pentru $\mathbf{X} \neq \mathbf{a}$, $S_{\bar{a}}(\mathbf{X}) = 0$ pentru $\mathbf{X} = \mathbf{a}$ și $S_{\bar{a}}(\mathbf{X}) = 1$ în caz contrar. $S_a(\mathbf{X}) + S_{\bar{a}}(\mathbf{X}) = 1$, $S_a(\mathbf{X})\, S_{\bar{a}}(\mathbf{X}) = 0$ pentru orice $\mathbf{X} \in \mathbf{B}_k^n$.

Lema. *Funcția booleană de n variabile polivalente* $f(x_1, x_2, \ldots, x_n)$ *se poate scrie sub forma* $f(\mathbf{X}) = \prod_{\mathbf{a} \in \mathbf{B}_k^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$.

Demonstrație. Considerăm expresia $f(\mathbf{X}) = \prod_{\mathbf{a} \in \mathbf{B}_k^n} [C(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$, $(\mathbf{a} = \mathbf{X} \Rightarrow S_{\bar{a}}(\mathbf{X}) = 0,\ \mathbf{a} \neq \mathbf{X} \Rightarrow S_{\bar{a}}(\mathbf{X}) = 1) \Rightarrow C(\mathbf{a}) = f(\mathbf{a})$.

Definiție. Expresia $f(\mathbf{X}) = \prod_{\mathbf{a} \in \mathbf{B}_k^n} [f(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$ a unei funcții booleene de n variabile polivalente se numește *forma normală canonică conjunctivă.*

Exemplu. Funcția $f(x_1, x_2) = x_2^0 (x_1^0 + x_1^2) + x_1^1 x_2^1$, în care variabilele x_1 și x_2 sînt trivalente, are următoarele forme normale canonice:

$$f(x_1, x_2) = x_1^0 x_2^0 + x_1^1 x_2^1 + x_1^2 x_2^0 =$$

$$= (x_1^1 + x_1^2 + x_2^0 + x_2^2)(x_1^1 + x_1^2 + x_2^0 + x_2^1)(x_1^0 + x_1^2 + x_2^1 + x_2^2) \times$$

$$\times (x_1^0 + x_1^2 + x_2^0 + x_2^1)(x_1^0 + x_1^1 + x_2^0 + x_2^2)(x_1^0 + x_1^1 + x_2^0 + x_2^1).$$

2.6.3. *Funcții de variabile polivalente inverse*

Definiție. Funcția de variabile polivalente $F(x_1, x_2, \ldots, x_n)$ este *inversa* funcției de variabile polivalente $f(x_1, x_2, \ldots, x_n)$, dacă îndeplinește condițiile

$$f(x_1, x_2, \ldots, x_n) + F(x_1, x_2, \ldots, x_n) = 1,$$

$$f(x_1, x_2, \ldots, x_n)\, F(x_1, x_2, \ldots, x_n) = 0.$$

Notăm prin $\bar{f}(x_1, x_2, \ldots, x_n)$ funcția booleană de variabile polivalente obținută din $f(x_1, x_2, \ldots, x_n)$ după următoarea regulă: semnele $+$ și $\cdot$ se permută între ele, constantele c se înlocuiesc cu $\bar{c}$, variabilele $x_i^{a_i}$ se înlocuiesc cu $x_i^{\bar{a}_i}$ pentru orice i.

Lema 1. *Dacă* $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a}\in\mathbf{B}_k^n} f(\mathbf{a})\, P_a(\mathbf{X})$, *atunci* $\bar{f}(x_1, x_2, \ldots, x_n) = \prod_{\mathbf{a}\in\mathbf{B}_k^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$.

Demonstrație. Aplicînd operatorul $^-$ funcției $f(x_1, x_2, \ldots, x_n)$, se obține succesiv: $\bar{f}(x_1, x_2, \ldots, x_n) = \prod_{\mathbf{a}\in\mathbf{B}_k^n} [\bar{f}(\mathbf{a}) + \bar{P}_a(\mathbf{X})] = \prod_{\mathbf{a}\in\mathbf{B}_k^n} [\bar{f}(\mathbf{a}) + S_{\bar{a}}(\mathbf{X})]$.

Într-adevăr

$$\bar{P}_a(\mathbf{X}) = \overline{x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}} = x_1^{\bar{a}_1} + x_2^{\bar{a}_2} + \ldots + x_n^{\bar{a}_n} = S_{\bar{a}}(\mathbf{X}).$$

Lema 2. *Funcția* $\bar{f}(x_1, x_2, \ldots, x_n)$ *este inversa funcției* $f(x_1, x_2, \ldots, x_n)$.

Demonstrație. Fie $f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a}\in\mathbf{B}_k^n} C_a P_a(\mathbf{X})$ și $\bar{f}(x_1, x_2, \ldots, x_n) = \prod_{\mathbf{a}\in\mathbf{B}_k^n} [\bar{C}_a + S_{\bar{a}}(\mathbf{X})]$. Atunci $\mathbf{X}=\mathbf{a} \Rightarrow f(\mathbf{X}) = \sum_{\mathbf{u}\in\mathbf{B}_k^n} C_u P_u(\mathbf{a}) = C_a$, $\bar{f}(\mathbf{X}) = \prod_{\mathbf{u}\in\mathbf{B}_k^n} [\bar{C}_a + S_{\bar{u}}(\mathbf{a})] = \bar{C}_a \Rightarrow f(\mathbf{X}) + \bar{f}(\mathbf{X}) = C_a + \bar{C}_a = 1$;

$\mathbf{X}=\bar{\mathbf{a}} \Rightarrow f(\mathbf{X}) = \sum_{\mathbf{u}\in\mathbf{B}_k^n} C_u P_u(\bar{\mathbf{a}}) = C_{\bar{a}}$, $\bar{f}(\mathbf{X}) = \prod_{\mathbf{u}\in\mathbf{B}_k^n} [\bar{C}_u + S_{\bar{u}}(\bar{\mathbf{a}})] = \bar{C}_{\bar{a}} \Rightarrow f(\mathbf{X}) + \bar{f}(\mathbf{X}) = 1$.

Apoi $C_a P_a(\mathbf{X})\,(\bar{C}_a + S_{\bar{a}}(\mathbf{X})) = C_a \bar{C}_a P_a(\mathbf{X}) + C_a P_a(\mathbf{X})\, S_{\bar{a}}(\mathbf{X}) = 0$ pentru orice $\mathbf{X}$ și a din $\mathbf{B}_k^n$ implică $f(\mathbf{X})\bar{f}(\mathbf{X}) = 0$.

Exemplu. Funcția inversă a funcției $f(x_1, x_2) = x_2^0(x_1^0 + x_1^2) + x_1^1 x_2^1$ are următoarele forme normale canonice:

$$\bar{f}(x_1, x_2) = (x_1^1 + x_1^2 + x_2^1 + x_2^2)\,(x_1^0 + x_1^2 + x_2^0 + x_2^2)\,(x_1^0 + x_1^1 + x_2^1 + x_2^2),$$

$$\bar{f}(x_1, x_2) = x_1^0 x_2^1 + x_1^0 x_2^2 + x_1^1 x_2^0 + x_1^1 x_2^2 + x_1^2 x_2^1 + x_1^2 x_2^2.$$

2.6.4. *Funcții de variabile polivalente incomplete*

Vom introduce și în acest caz noțiunea de *funcție de variabile polivalente incompletă* ca în cazul bivalent.

Definiție. O funcție de variabile polivalente pentru care unii coeficienți ai formelor normale canonice pot lua valori arbitrare, 0 sau 1, se numesc *funcții de variabile polivalente incomplete*.

Dacă f_0 este funcția obținută din f prin egalarea cu 0 a tuturor coeficienților nedefiniți și f_1 funcția obținută prin egalarea cu 1 a tuturor coeficienților nedefiniți din f, atunci $f_0 \Rightarrow f_1$. Această relație va fi notată ca și în cazul bivalent $f_0 \leqslant f_1$.

Funcția f incomplet definită va fi notată $f = (f_0, f_1)$ și va reprezenta mulțimea funcțiilor g cu proprietatea $f_0 \leqslant g \leqslant f_1$.

Definiție. Ca și în cazul bivalent vom defini *inversa* unei funcții incomplete de variabile polivalente (f_0, f_1) prin perechea $(\bar{f}_1, \bar{f}_0)$.

Exemplu. Inversa funcției (f_0, f_1), unde $f_0 = x_1^0 x_2^0 + x_1^1 x_2^1 + x_1^2 x_2^0$, $f_1 = x_1^0 x_2^0 + x_1^1 x_2^0 + x_1^1 x_2^1 + x_1^1 x_2^2 + x_1^2 x_2^0 = x_1^1 + x_2^0$, este funcția $(\bar{f}_1, \bar{f}_0)$, unde $\bar{f}_1 = x_1^0 x_2^1 + x_1^0 x_2^2 + x_1^2 x_2^1 + x_1^2 x_2^2 = (x_1^0 + x_1^2)\,(x_2^1 + x_2^2)$, $\bar{f}_0 = x_1^0 x_2^1 + x_1^0 x_2^2 + x_1^1 x_2^0 + x_1^1 x_2^2 + x_1^2 x_2^1 + x_1^2 x_2^2 = x_1^0 x_2^1 + x_1^1 x_2^0 + x_1^2 x_2^1 + x_1^2 x_2^2$.

Prin inversarea unei funcții de variabile polivalente incomplete se permută ordinea termenilor, ca și în cazul bivalent: $f_0 \leqslant f_1 \Rightarrow \bar{f}_1 \leqslant \bar{f}_0$.

2.6.5. *Semilaticea funcțiilor de variabile polivalente incomplete*

Ca și în cazul bivalent se poate demonstra că mulțimea funcțiilor de variabile polivalente poate fi organizată ca semilatice față de adunare, cu ajutorul relației

$$f \leqslant g \Leftrightarrow (g_0 \leqslant f_0, \quad f_1 \leqslant g_1),$$

unde $f = (f_0, f_1)$ și $g = (g_0, g_1)$ sînt funcții de variabile polivalente incomplete.

Întrucît toate rezultatele obținute în 2.5.2 sînt valabile și pentru funcțiile de variabile polivalente și demonstrațiile se fac la fel ca în 2.5.2 acestea nu se vor mai repeta aici.

2.6.6. *Algebra booleană a funcțiilor de variabile polivalente*

După cum s-a arătat, funcțiile de variabile polivalente iau numai valorile 0 și 1 și admit inverse. Ele pot fi considerate pe de o parte algebre booleene de variabilele $x_i^{a_i}$, $i \in \mathbf{I}_n$, $a_i \in \{0, \ldots, k_i - 1\}$, între care există legăturile: $\sum_{j \in \{0,\ldots,k_i-1\}} x_i^j = 1$, $x_i^u x_i^v = 0$, $u \neq v$, $i \in \mathbf{I}_n$, iar pe de altă parte algebre de n variabile polivalente, adică extinderi ale algebrelor booleene de n variabile.

CAPITOLUL 3

ECUAȚII BOOLEENE. ECUAȚII DE VARIABILE POLIVALENTE

3.1. Egalități booleene

Problema ecuațiilor booleene este rar tratată în lucrările de teoria automatelor. Vom prezenta această problemă mai dezvoltat, întrucît rezolvarea unor ecuații booleene intervine în algoritmul de sinteză dat de autor, descris în cap. 13.

Proprietatea 1. *Orice ecuație booleană $f = g$ poate fi scrisă sub forma $fg + \bar{f}\bar{g} = 0$.*

Proprietatea 2. *Sistemul de două ecuații booleene $f = 0$, $g = 0$ este echivalent cu ecuația unică $f + g = 0$.*

Proprietatea 2 poate fi extinsă imediat la cazul general. Sistemul de n ecuații booleene $f_i = 0$, $i = 1, 2, \ldots, n$, este echivalent cu ecuația unică $\sum_{i=1}^{n} f_i = 0$.

3.2. Ecuații booleene de o variabilă

Să considerăm ecuația booleană $\bar{x} f_1 + x f_2 = 0$, unde f_1 și f_2 nu depind de x.

Teoremă. *Ecuația* $\bar{x}f_1 + xf_2 = 0$ *admite soluția* $x = (f_1, \bar{f}_2)$.

Demonstrație. Conform proprietății 2 această ecuație este echivalentă cu sistemul de ecuații : $\bar{x}f_1 = 0$, $xf_2 = 0$, de unde se obține imediat, folosind proprietățile implicației,

$$f_1 \leqslant x \leqslant \bar{f}_2 \Rightarrow x = (f_1, \bar{f}_2).$$

Consecință. *Rezultă că ecuația* $\bar{x}f_1 + xf_2$ *are soluția numai dacă este îndeplinită condiția* $f_1 \leqslant \bar{f}_2$ *care este echivalentă cu condiția* $f_1 f_2 = 0$.

Am arătat astfel că soluția ecuației booleene $\bar{x}f_1 + xf_2 = 0$ este funcție incompletă $(f_1, \bar{f}_2)$. Se poate demonstra și reciproc că orice funcție incompletă este soluție a unei ecuații booleene, respectiv se poate stabili o corespondență biunivocă între mulțimea funcțiilor incomplete și mulțimea soluțiilor ecuațiilor booleene.

Mulțimea funcțiilor incomplete poate fi considerată ca o extindere algebrică a mulțimii funcțiilor booleene, în sensul în care mulțimea numerelor algebrice este o extindere a mulțimii numerelor raționale.

Corespondența stabilită între mulțimea soluțiilor ecuațiilor booleene și mulțimea funcțiilor incomplete reduce problema rezolvării ecuațiilor booleene la studiul funcțiilor incomplete asociate, ceea ce se va dovedi foarte util. Aceasta ne va permite să dăm o metodă de rezolvare a ecuațiilor booleene, pe care o vom numi metoda substituției.

3.3. Metoda substituției

Să considerăm ecuația $f(x_1, x_2, \ldots, x_n) = 0$, unde $f(x_1, x_2, \ldots, x_n)$ este o funcție booleană arbitrară de variabilele $x_1, x_2, \ldots, x_n$. Ecuația se mai poate scrie

$$\bar{x}_1 f(0, x_2, \ldots, x_n) + x_1 f(1, x_2, \ldots, x_n) = 0$$

și, ținînd seama de teorema din 3.2, admite soluția

$x_1 = (f(0, x_2, \ldots, x_n), \bar{f}(1, x_2, \ldots, x_n))$ cu condiția de existență $f(0, x_2, \ldots, x_n) f(1, x_2, \ldots, x_n)) = 0$.

În acest paragraf pentru simplificarea scrierii vom folosi notațiile $\mathbf{A}_{ij} = (a_i, a_{i+1}, \ldots, a_j)$, $\mathbf{X}_{ij} = (x_i, x_{i+1}, \ldots, x_j)$.

Lema 1. *Variabila* x_k *se obține în funcție de variabilele* $x_{k+1}, x_{k+2}, \ldots, x_n$ *și parametrii booleeni arbitrari* $a_1, a_2, \ldots, a_{k+1}$, *după formula* :

$$x_k = \Big(\prod_{\mathbf{A}_{1,k-1} \in \mathbf{B}_2^{k-1}} f(\mathbf{A}_{1,k-1}, 0, \mathbf{X}_{k+1,n}), \sum_{\mathbf{A}_{1,k-1} \in \mathbf{B}_2^{k-1}} \bar{f}(\mathbf{A}_{1,k-1}, 1, \mathbf{X}_{k+1}) \Big)$$

cu condiția $\prod_{\mathbf{A}_{1,k} \in \mathbf{B}_2^k} f(\mathbf{A}_{1,k}\ \mathbf{X}_{k+1,n}) = 0.$

Demonstrație. Lema este adevărată pentru $k = 1$, după cum s-a arătat mai înainte. Mai departe facem demonstrația prin metoda inducției complete după k. Presupunem deci lema adevărată pentru $k - 1$. Atunci

$$x_{k-1} = \Big(\prod_{\mathbf{A}_{1,k-2} \in \mathbf{B}_2^{k-2}} f(\mathbf{A}_{1,k-2}, 0, \mathbf{X}_{k,n}), \sum_{\mathbf{A}_{1,k-2} \in \mathbf{B}_2^{k-2}} \bar{f}(\mathbf{A}_{1,k-2}, 1, \mathbf{X}_{k,n}) \Big)$$

cu condiția $\prod_{\mathbf{A}_{1,k-1} \in \mathbf{B}_2^{k-1}} f(\mathbf{A}_{1,k-1}, \mathbf{X}_{k,n}) = 0$. Dar această ultimă ecuație admite soluția din enunțul lemei, ceea ce trebuia demonstrat.

Deoarece k a fost ales arbitrar, lema 1 este adevărată pentru orice $k \in [1, n]$. Pentru x_{n-1} vom obține soluția

$$x_{n-1} = \Big(\prod_{\mathbf{A}_{1,n-2} \in \mathbf{B}_2^{n-2}} f(\mathbf{A}_{1,n-2}, 0, x_n), \sum_{\mathbf{A}_{1,n-2} \in \mathbf{B}_2^{n-2}} \bar{f}(\mathbf{A}_{1,n-2}, 1, x_n) \Big)$$

cu condiția $\prod_{\mathbf{A}_{1,n-1} \in \mathbf{B}_2^{n-1}} f(\mathbf{A}_{1,n-1}, x_n) = 0$ care conduce la următoarea soluție pentru x_n :

$$x_n = \Big(\prod_{\mathbf{A}_{1,n-1} \in \mathbf{B}_2^{n-1}} f(\mathbf{A}_{1,n-1}, 0), \sum_{\mathbf{A}_{1,n-1} \in \mathbf{B}_2^{n-1}} \bar{f}(\mathbf{A}_{1,n-1}, 1) \Big)$$

cu condiția $\prod_{\mathbf{A}_{1,n}\in\mathbf{B}_2^n} f(\mathbf{A}_{1,n}) = 0$, unde $\mathrm{A}_{1,n} = (a_1, a_2, \ldots, a_n)$ este un vector boolean cu n componente. Consecința imediată a acestui rezultat este :

Lema 2. *Condiția de existență a soluției ecuației booleene de n variabile $f(x_1, x_2, \ldots, x_n) = 0$ este* $\prod_{\mathbf{A}_{1,n}\in\mathbf{B}_2^n} f(\mathbf{A}_{1,n}) = 0$.

Se poate obține o soluție recursivă a ecuației booleene de n variabile.

Teorema 1. *Ecuația booleană de n variabile $f(x_1, x_2, \ldots, x_n) = 0$ admite următoarea soluție recursivă :*

$$\begin{aligned}
x_1 &= \Big(\prod_{\mathbf{A}_{2,n}\in\mathbf{B}_2^{n-1}} f(0, \mathbf{A}_{2,n}),\ \sum_{\mathbf{A}_{2,n}\in\mathbf{B}_2^{n-1}} \bar{f}(1, \mathbf{A}_{2,n})\Big),\\
&\vdots\\
x_k &= \Big(\prod_{\mathbf{A}_{k+1,n}\in\mathbf{B}_2^{n-k}} f(\mathbf{X}_{1,k-1}, 0, \mathbf{A}_{k+1,n}),\\
&\qquad \sum_{\mathbf{A}_{k+1,n}\in\mathbf{B}_2^{n-k}} \bar{f}(\mathbf{X}_{1,k-1}, 1, \mathbf{A}_{k+1,n})\Big),\\
x_n &= (f(\mathbf{X}_{1,n-1}, 0),\ \bar{f}(\mathbf{X}_{1,n-1}, 1)).
\end{aligned}$$

Demonstrație. Schimbînd rolul lui x_n cu x_1 și folosind lema 1, se obține x_1. Apoi, schimbînd rolul lui x_{n-1} cu x_2 și ținînd seama de lema 1, se obține x_2. Urmînd același procedeu, obținem pe x_k, pentru orice $k \in [1, n]$.

Observație. Funcția incompletă $f = (f_0, f_1)$ se poate scrie și sub forma $f = f_0 + u f_1$, $u \in (0, 1)$.

Pe baza acestei observații se obține următoarea consecință a teoremei 1.

Teorema 2. *Soluția ecuației booleene de n variabile $f(x_1, x_2, \ldots, x_n) = 0$ se poate scrie sub următoarea formă parametrică, recursivă :*

$$\begin{aligned}
x_1 &= \prod_{\mathbf{A}_{2,n}\in\mathbf{B}_2^{n-1}} f(0, \mathbf{A}_{2,n}) + u_1 \sum_{\mathbf{A}_{2,n}\in\mathbf{B}_2^{n-1}} \bar{f}(1, \mathbf{A}_{2,n}),\\
&\vdots\\
x_k &= \prod_{\mathbf{A}_{k+1,n}\in\mathbf{B}_2^{n-k}} f(\mathbf{X}_{1,n-1}, 0, \mathbf{A}_{k+1,n}) +\\
&\qquad + u_k \sum_{\mathbf{A}_{k+1,n}\in\mathbf{B}_2^{n-k}} \bar{f}(\mathbf{X}_{1,n-1}, 1, \mathbf{A}_{k+1,n}),\\
&\vdots\\
x_n &= f(\mathbf{X}_{1,n}, 0) + u_n \bar{f}(\mathbf{X}_{1,n}, 1),\ u_i \in (0, 1),\ (\forall) i \in \mathbf{I}_n.
\end{aligned}$$

Exemplu. Să se rezolve parametric ecuația $\bar{f}(x,y,z) \equiv xy + yz + zx = 0$. Aplicînd teorema 2, se obține succesiv : $z = [f(x, y, 0) + u_3 \bar{f}(x, y, 1) = xy + \bar{x}\bar{y}\, u_3$, $y = f(x, 0, 0) f(x, 0, 1) + u_2 [\bar{f}(x, 1, 0) + \bar{f}(x, 1, 1)] = \bar{x}\, u_2$, $x = f(0,0,0) f(0,0,1) f(0,1,0) f(0,1,1) + u_1 [f(1,0,0) + f(1,0,1) + f(1,1,0) + f(1,1,1)] = u_1$ și apoi, făcînd substituțiile, se obține soluția $x = u_1$, $y = \bar{u}_1 u_2$, $z = \bar{u}_1 \bar{u}_2 u_3$.

Soluția se obține la fel de repede aplicînd teorema 1. Într-adevăr, se deduce succesiv $x = (yz, \overline{y\bar{z}}) = (0, \bar{u}, \bar{z}) = v\bar{u}\bar{z}$, $y = (0, \bar{z}) = u\bar{z}$, z arbitrar.

Deci în practică ecuația dată se înlocuiește cu un sistem de n ecuații cu n necunoscute, eliminînd succesiv cîte o variabilă. Metoda substituției este justificată de

Teorema 3. *Ecuația booleană $f(x_1, x_2, \ldots, x_n) = 0$ este echivalentă cu următorul sistem de n ecuații cu n necunoscute : $f_k(x_k, \ldots, x_n) = 0$, $k = 1, 2, \ldots, n$, unde*

$$f_1(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n),$$

$$f_k(x_k, x_{k+1}, \ldots, x_n) = \prod_{\mathbf{A}_{1,k-1}\in\mathbf{B}_2^{k-1}} f(\mathbf{A}_{1,k-1}, \mathbf{X}_{k,n}),\ k \in \{2, \ldots, n\}.$$

Demonstrația este imediată, întrucît sistemul precedent este constituit de ecuația însăși și de condițiile de existență a soluției, deduse anterior.

Exemplu. Să se rezolve prin metoda substituției ecuația $ax + by + cz = 0$.

Sistemul asociat va fi

$$f_1(x, y, z) \equiv ax + by + cz = 0,$$

$$f_2(y, z) \equiv by + cz = 0,$$

$$f_3(z) \equiv cz = 0,$$

de unde se obține soluția $z = \bar{c}u_3$, $y = \bar{b}u_2$, $x = \bar{a}u_1$.

3.4. Metoda lui Löwenheim

O altă metodă de rezolvare a ecuațiilor booleene a fost dată de L. Löwenheim [42]. Metoda lui Löwenheim dă soluția generală a ecuației booleene $f(x_1, x_2, \ldots, x_n) = 0$ în funcție de o soluție particulară $x_i = a_i$, $i = 1,2, \ldots, n$, și de n parametri arbitrari u_i, $i = 1, 2, \ldots, n$. În [42] s-a arătat că dacă ecuația este rezolvabilă, atunci ea admite soluția particulară

$$a_i = \sum_{(\mathbf{B}_{1,i-1}, \mathbf{B}_{i+1,n}) \in \mathbf{B}_2^{n-1}} \bar{f}(\mathbf{B}_{1,i-1}, 1, \mathbf{B}_{i+1,n}) \times$$

$$\times \prod_{(c_1, c_2, \ldots, c_n) > (\mathbf{B}_{1,i-1}, 1, \mathbf{B}_{i+1,n})} f(c_1, c_2, \ldots, c_n),$$

$$\mathbf{B}_{1,i-1} = (b_1, b_2, \ldots, b_{i-1}), \quad \mathbf{B}_{i+1,n} = (b_{i+1}, \ldots, b_n),$$

deci ecuația admite soluția generală parametrică

$$x_i = f(u_1, \ldots, u_n) \sum_{(\mathbf{B}_{1,i-1}, \mathbf{B}_{i+1,n}) \in \mathbf{B}_2^{n-1}} \bar{f}(\mathbf{B}_{1,i-1}, 1, \mathbf{B}_{i+1,n}) \times$$

$$\times \prod_{(c_1, c_2, \ldots, c_n) > (\mathbf{B}_{1,i-1}, 1, \mathbf{B}_{i+1,n})} f(c_1, \ldots, c_n), + u_i \bar{f}(u_1, u_2, \ldots, u_n).$$

Demonstrație. Evident o soluție a ecuației $f(x_1, x_2, \ldots, x_n) = 0$, $(t_1, t_2, \ldots, t_n)$, se bucură de proprietatea A: $f(t_1, t_2, \ldots, t_n) = 0, \bar{f}(t_1, t_2, \ldots, t_n) = 1$. Reciproc, orice sistem de valori ale necunoscutelor $(s_1, s_2, \ldots, s_n)$ care nu este o soluție a ecuației se bucură de proprietatea B:

$$f(s_1, s_2, \ldots, s_n) = 1,$$

$$\bar{f}(s_1, s_2, \ldots, s_n) = 0.$$

Mulțimea șirurilor de lungime n, formate din 0 și 1, se împarte deci în două clase pe care le notăm $\mathbf{I}$ și $\mathbf{J}$, cu proprietățile :

$$(t_1, t_2, \ldots, t_n) \in \mathbf{I} \Rightarrow A,$$

$$(s_1, s_2, \ldots, s_n) \in \mathbf{J} \Rightarrow B.$$

Evident orice șir de lungime n, format din 0 și 1, este conținut fie în $\mathbf{I}$, fie în $\mathbf{J}$.

Fie acum o soluție particulară $(a_1, a_2, \ldots, a_n)$ și un șir arbitrar $(u_1, u_2, \ldots, u_n)$. Dacă $(u_1, u_2, \ldots, u_n) \in \mathbf{I}$, atunci $(u_1 \bar{f}(u_1, \ldots, u_n), u_2 \bar{f}(u_1, \ldots, u_n), \ldots, u_n \bar{f}(u_1, \ldots, u_n)) = (u_1, u_2, \ldots, u_n)$ este soluție a ecuației. Dacă $(u_1, u_2, \ldots, u_n) \in \mathbf{J}$, atunci $(a_1 f(u_1, \ldots, u_n), a_2 f(u_1, \ldots, u_n), \ldots, a_n f(u_1, \ldots, u_n)) = (a_1, a_2, \ldots, a_n)$ este conform presupunerii soluție a ecuației.

Ținînd seama de cele de mai înainte și de faptul că $\mathbf{I} \cup \mathbf{J}$ acoperă întreaga mulțime a șirurilor de lungime n, se deduce că

$x_i = a_i f(u_1, \ldots, u_n) + u_i \bar{f}(u_1, \ldots, u_n)$, $i \in \{1, \ldots, n\}$, este soluție a ecuației, ceea ce trebuia demonstrat.

3.5. Ecuații de variabile polivalente

3.5.1. *Ecuații de o variabilă polivalentă*

Să considerăm ecuația de o variabilă polivalentă $\sum_{i=1}^{k} x^{a_i} f_i = 0$, în care $\sum_{i=1}^{k} x^{a_i} = 1$, $x^{a_i} x^{a_j} = 0$, $i \neq j$.

Teoremă. *Ecuația de o variabilă polivalentă* $\sum_{i=1}^{k} x^{a_i} f_i = 0$ *are ca soluție următoarele* k *funcții* : $x^{a_i} = (\prod_{j \neq i} f_j, \bar{f}_i)$, $i \in \{1, \ldots, k\}$, *de variabile polivalente incomplete, cu condiția de rezolvabilitate* $\prod_{i=1}^{k} f_i = 0$.

Demonstrație. Întrucît x^{a_i} și f_i, $i = 1, 2, \ldots, k$, sînt variabile booleene, ecuația poate fi considerată ecuație booleană și este echivalentă cu sistemul de ecuații $x^{a_i} f_i = 0$, $i = 1, 2, \ldots, k$, care sînt echivalente cu relațiile $x^{a_i} \leqslant \bar{f}_i$, $i = 1, 2, \ldots, n$. Însumînd după i, se obține $\sum_{i=1}^{k} x^{a_i} \leqslant \sum_{i=1}^{k} \bar{f}_i$. Ținînd seama că $\sum_{i=1}^{k} x^{a_i} = 1$, rezultă $\sum_{i=1}^{k} \bar{f}_i = 1$

sau $\prod_{i=1}^{k} f_i = 0$, care reprezintă condiția de rezolvabilitate a ecuației, similară cu cea din cazul bivalent. Pe de altă parte, din relația $x^{\bar{a}i} = \sum_{a_j \neq a_i} x^{a_j}$ și din $x^{a_i} \leqslant \bar{f}_i$ se deduce $x^{\bar{a}i} \leqslant \sum_{j \neq i} \bar{f}_j$ și $x^{a_i} \geqslant \prod_{j \neq i} f_j$, de unde rezultă că x^{a_i} satisface dubla inegalitate $\prod_{j \neq i} f_j \leqslant x^{a_i} \leqslant \bar{f}_i$, $(\forall)\ i \in \{1, \ldots, k\}$, adică $x^{a_i} = (\prod_{j \neq i} f_j,\ f_i)$, ceea ce trebuia demonstrat.

Exemplu. Fie ecuația $x^0 + x^2 = 0$, unde x este o variabilă trivalentă. Avem $f_1 = 1$, $f_2 = 0$, $f_3 = 1$. Condiția de rezolvabilitate fiind îndeplinită, aplicăm teorema și se obține $x^0 = (f_2 f_3, \bar{f}_1) = (0,\ 0) = 0$, $x^1 = (f_1 f_3,\ f_2) = (1,\ 1) = 1$, $x^2 = (f_1 f_2,\ f_3) = (0,\ 0) = 0$, adică $x = 1$.

3.5.2. *Metoda substituției*

Să considerăm ecuația de variabile polivalente $f(x_1, x_2, \ldots, x_n) = 0$, unde $f(x_1, x_2, \ldots, x_n)$ este o funcție arbitrară de n variabile polivalente.

Teoremă. *Ecuația de n variabile polivalente $f(x_1, x_2, \ldots, x_n) = 0$ admite soluția recursivă*

$$x_1^{a_{1i}} = \prod_{\substack{a_1 \in \mathbf{B}_k^1 \\ a_1 \neq a_{1i}}} f(a_1,\ x_2,\ \ldots, x_n),\ \bar{f}(a_{1i}, x_2, \ldots, x_n)),$$

$$\vdots$$

$$x_j^{a_{ji}} = (\prod_{\substack{\mathbf{A}_{1j} \in \mathbf{B}_k^j \\ a_j \neq a_{ji}}} f(\mathbf{A}_{1,j},\ \mathbf{X}_{j+1,n}),\ \sum_{\mathbf{A}_{1,j-1} \in \mathbf{B}_k^{j-1}} \bar{f}(\mathbf{A}_{1,j-1}, a_{ji}, \mathbf{X}_{j+1,n})),$$

$$\vdots$$

$$x_n^{a_{ni}} = (\prod_{\substack{\mathbf{A}_{1,n} \in \mathbf{B}_k^n \\ a_n \neq a_{ni}}} f(\mathbf{A}_{1,n}),\ \sum_{\mathbf{A}_{1,n-1} \in \mathbf{B}_k^{n-1}} \bar{f}(\mathbf{A}_{1,n-1},\ a_{n,i})),$$

$$i \in \{1, \ldots, k_j\}, \qquad (\forall)\ j \in \{1, \ldots, n\},$$

cu condiția de rezolvabilitate $\prod_{\mathbf{A}_{1,n} \in \mathbf{B}_k^n} f(\mathbf{A}_{1,n}) = 0.$

Demonstrație. Dezvoltînd după variabila x_1, se poate scrie

$$\sum_{i=1}^{k_1} x_1^{a_{1,i}} f(a_{1,i}, x_2, \ldots, x_n) = 0.$$

Ținînd seama de teorema din 3.5.1, rezultă

$$x_1^{a_{1i}} = (\prod_{a_1 \neq a_{1i}} f(a_1,\ x_2, \ldots, x_n),\ \bar{f}(a_{1i},\ x_2,\ \ldots,\ x_n)),$$

$$i = 1, 2, \ldots, k_i,$$

cu condiția $\prod_{i=1}^{k_i} f(a_{1i}, x_2, \ldots, x_n) = 0$, care este o ecuație de variabilele $x_2, x_3, \ldots, x_n$. Rezolvînd ecuația precedentă în raport cu x_2, vom obține valoarea lui x_2 în raport cu $x_3, x_4, \ldots, x_n$ și doi parametri a_1 și a_2. În general, vom obține pe x_k în raport cu $x_{k+1}, x_{k+2}, \ldots, x_n$ și k parametri $a_1, a_2, \ldots, a_k$ sub forma

$$x_j^{a_{ji}} = (\prod_{\substack{\mathbf{A}_{1,j} \in \mathbf{B}_k^j \\ a_j \neq a_{ji}}} f(\mathbf{A}_{1,j},\ \mathbf{X}_{j+1,n}),\ \sum_{\mathbf{A}_{1,j-1} \in \mathbf{B}_k^{j-1}} \bar{f}(\mathbf{A}_{1,j-1},\ a_{ji},\ \mathbf{X}_{j+1,n}))$$

cu condiția de rezolvabilitate $\prod_{\mathbf{A}_{1,j} \in \mathbf{B}_k^j} f(\mathbf{A}_{1,j},\ \mathbf{X}_{j+1,n}) = 0$. În fine,

$$x_n^{a_{ni}} = (\prod_{\substack{\mathbf{A}_{1,n} \in \mathbf{B}_k^n \\ a_n \neq a_{ni}}} f(\mathbf{A}_{1,n}),\ \sum_{\mathbf{A}_{1,n-1} \in \mathbf{B}_k^{n-1}} \bar{f}(\mathbf{A}_{1,n-1},\ a_{ni}))$$

cu condiția de rezolubilitate $\prod_{\mathbf{A}_{1,n} \in \mathbf{B}_k^n} f(\mathbf{A}_{1,n}) = 0$, unde ca de obicei prin $\mathbf{A}_{ij}$ am notat vectorul $\mathbf{A}_{i,j} = (a_i,\ a_{i+1}, \ldots, a_j)$. Cu aceasta, teorema este complet demonstrată.

Exemplu. Să se rezolve $f(x_1, x_2, x_3) = (x_1^0 + x_1^2)(x_2^0 + x_2^2)(x_3^0 + x_3^2) + x_1^1 x_2^1 x_3^1 = 0$. Rezultă succesiv: $f(0, x_2, x_3) = (x_2^0 + x_2^2)(x_3^0 + x_3^2)$, $f(1, x_2, x_3) = x_2^1 x_3^1$, $f(2, x_2, x_3) = (x_2^0 + x_2^2)(x_3^0 + x_3^2)$, $\bar{f}(0, x_2, x_3) = x_2^1 + x_3^1$, $\bar{f}(1, x_2, x_3) = (x_2^0 + x_2^2)(x_3^0 + x_3^2)$, $\bar{f}(2, x_2, x_3) = x_2^1 + x_3^1$, și aplicînd teoremă rezultă x_1 astfel: $x_1^0 = (0, x_2^1 + x_3^1)$, $x_1^1 = ((x_2^0 + x_2^2)(x_3^0 + x_3^2), x_2^0 + x_2^2 + x_3^0 + x_3^2)$, $x_1^2 = (0, x_2^1 + x_3^1)$.

Întrucît condiția $f(0, x_2, x_3) f(1, x_2, x_3) f(2, x_2, x_3) = (x_2^0 + x_2^2)(x_3^0 + x_3^2) x_2^1 x_3^1 (x_2^0 + x_2^2)(x_3^0 + x_3^2) = 0$

este o identitate, variabilele x_2 și x_3 pot fi luate arbitrar.

Luînd pentru x_2 și x_3 valori arbitrare, rezultă 18 soluții ale ecuației date (vezi tabelul 3.1).

Tabelul 3.1

Numărul soluției	x_2	x_3	x_1^0	x_1^1	x_1^2	x_1
1	0	0	0	1	0	1
2	0	1	1	0	0	0
3	0	1	0	1	0	1
4	0	1	0	0	1	2
5	0	2	0	1	0	1
6	1	0	1	0	0	0
7	1	0	0	1	0	1
8	1	0	0	0	1	2
9	1	1	1	0	0	0
10	1	1	0	0	1	2
11	1	2	1	0	0	0
12	1	2	0	1	0	1
13	1	2	0	0	1	2
14	2	0	0	1	0	1
15	2	1	1	0	0	0
16	2	1	0	1	0	1
17	2	1	0	0	1	2
18	2	2	0	1	0	1

3.5.3. *Metoda lui Löwenheim extinsă la ecuații de variabile polivalente*

Vom extinde utilizarea metodei lui Löwenheim la ecuațiile de variabile polivalente.

Fie ecuația $f(x_1, x_2, \ldots, x_n) = 0$, unde $f(x_1, x_2, \ldots, x_n)$ este o funcție de variabile polivalente.

Teoremă. *Dacă* $(a_1, a_2, \ldots, a_n)$ *este o soluție particulară a ecuației și dacă* $(u_1, u_2, \ldots, u_n)$ *este un vector arbitrar de variabile polivalente, atunci soluția generală a ecuației se poate scrie astfel :*

$$x_i = a_i f(u_1, u_2, \ldots, u_n) + u_i \bar{f}(u_1, u_2, \ldots, u_n),$$

$$i \in \{1, \ldots, n\}.$$

Demonstrația este similară cu cea pentru cazul bivalent dată în 3.4.

CODURI ASOCIATE FUNCȚIILOR BOOLEENE ȘI FUNCȚIILOR DE VARIABILE POLIVALENTE

4.1. Generalități

În studiul automatelor finite, teoria codurilor intervine sub forme și cu funcțiuni multiple. Prima problemă care se pune este codificarea informațiilor care se introduc în automat și care se obțin la ieșirea acestuia. Omul furnizează informațiile și dorește să-i fie furnizate sub formă alfanumerică, folosind pentru informațiile numerice sistemul zecimal iar pentru informațiile nenumerice, literele, în timp ce dispozitivele tehnologice sînt construite, în general, pentru prelucrarea binară a informației. Din această cauză a fost necesar să se construiască pe de o parte dispozitive care să transforme informațiile alfanumerice în informații binare, realizînd ceea ce se numește codificarea informației alfanumerice, iar pe de altă parte dispozitive care să transforme informațiile binare în informații alfanumerice, realizînd decodificarea informației. S-au creat în acest scop codurile zecimale și codurile alfanumerice.

A doua problemă în care intervine teoria codurilor este transmiterea informației. În general, informațiile nu pot fi transmise la distanță sub forma sub care au fost produse, ci trebuie codificate. Ele suferă deci trei operații : codificarea, transmiterea și decodificarea. În cursul transmisiei informațiile sînt supuse perturbațiilor, existînd riscul de apariție a unor erori. Din această cauză, a fost necesar să

se creeze mijloacele de detectare și, eventual, de corectare a erorilor. Această preocupare a condus la crearea codurilor detectoare și a codurilor corectoare de erori, dintre care amintim codurile Hamming.

Nu vom insista mai mult asupra primelor două probleme care au solicitat utilizarea codurilor, în studiul automatelor finite, întrucît aceste probleme nu fac obiectul acestei cărți. În continuare, vom dezvolta, în schimb, teoria codurilor, legată de reprezentarea funcțiilor booleene și funcțiilor de variabile polivalente și prin aceasta de analiza și sinteza automatelor.

4.2. Coduri bivalente

4.2.1. *Codificarea binară a mulțimilor de elemente*

Fie $\mathbf{O} = (o_1, o_2, \ldots, o_m)$ o mulțime de obiecte. Să considerăm mulțimea vectorilor bivalenți cu n componente, $\mathbf{B}_2^n$, unde n verifică relația $2^n \geqslant m$. Fie $\mathbf{K} \subset \mathbf{B}_2^n$, o parte a lui $\mathbf{B}_2^n$, care conține m vectori $\mathbf{a}_i = (a_{i,1}, a_{i,2}, \ldots, a_{in})$, $i = 1,2, \ldots, m$.

Definiție. Se spune că mulțimea $\mathbf{O}$ a fost *codificată binar* sau că i s-a asociat un cod bivalent dacă s-a stabilit o corespondență biunivocă între mulțimea $\mathbf{O}$ și o parte $\mathbf{K} \subset \mathbf{B}_2^n$.

4.2.2. *Coduri bivalente ponderate*

În cazurile practice se urmărește stabilirea corespondenței dintre mulțimile codificate și coduri după anumite reguli. O astfel de regulă se utilizează în codurile ponderate.

Fie $\mathbf{K}$ un cod bivalent, de n componente, sau, ceea ce este același lucru, o matrice $\mathbf{K}$ de vectori din $\mathbf{B}_2^n$. Dacă fiecărei coloane j din matricea $\mathbf{K}$ i se asociază o pondere Q_j, atunci fiecărei linii $\mathbf{a}_i$ i se asociază un număr p_i dat de expresia $p_i = \sum_{j=1}^{n} a_{ij} Q_j$.

Exemplu. Codul *DCBN* (zecimal codificat binar natural) pentru numerele de la 0 la 9 este cod binar ponderat cu ponderile $q_1 = 8$, $q_2 = 4$, $q_3 = 2$ și $q_4 = 1$, după cum se poate verifica în tabelul 4.1.

Tabelul 4.1

p_i	8	4	2	1	p_i	8	4	2	1
0	0	0	0	0	5	0	1	0	1
1	0	0	0	1	6	0	1	1	0
2	0	0	1	0	7	0	1	1	1
3	0	0	1	1	8	1	0	0	0
4	0	1	0	0	9	1	0	0	1

Observație. Codul *DCBN* se spune că este cod binar natural, întrucît componentele a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4} ale codului asociat oricărei cifre zecimale a sînt cifrele din scrierea în baza 2, $a_{i_1} a_{i_2} a_{i_3} a_{i_4}$ a lui a_i.

4.2.3. *Coduri grupale*

Fie $\mathbf{K}$ un cod bivalent cu matricea de cod $\mathbf{K} = (a_{ij})$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$.

Definiția 1. Dacă a_i și a_k sînt două linii arbitrare din $\mathbf{K}$, între aceste linii se introduce operația $\oplus$, dată de expresia $a_i \oplus a_k = (a_{i1} \oplus a_{k1}, a_{i2} \oplus a_{k2}, \ldots, a_{in} \oplus a_{kn})$, unde $a_{ij} \oplus a_{kj}$ este suma modulo 2 a variabilelor booleene a_{ij} și a_{kj}, $j = 1,2, \ldots, n$. Operația $\oplus$ va fi denumită *suma modulo* 2 a vectorilor a_i și a_k.

Definiția 2. Un cod $\mathbf{K}$ se numește *cod grupal*, dacă mulțimea liniilor lui $\mathbf{K}$ formează grup față de operația $\oplus$.

Să considerăm expresia $d_{ik} = \sum_{j=1}^{n} a_{ij} \oplus a_{kj}$ denumită *distanța Hamming* între liniile a_i și a_k. Distanța Hamming joacă rol în detectarea și corectarea erorilor. Se demonstrează că un cod pentru care distanța Hamming minimă este d, detectează $d-1$ erori. Mai mult, între numărul de detectări p, numărul de corectări q și distanța Hamming minimă există relațiile

$$p + q = d - 1, \quad p \geqslant q.$$

Un exemplu de cod grupal este codul $\mathbf{B}_2^n$, pentru orice n.

Într-adevăr, fie $\theta = (0, 0, \ldots, 0) \in \mathbf{B}_2^n$ vectorul cu toate componentele nule. Se verifică imediat relațiile :

$$a \oplus b \in \mathbf{B}_2^n, \ (\forall)\ a, b \in \mathrm{B}_2^n,$$

$$a \oplus \theta = a, \quad a \oplus a = \theta,$$

de unde rezultă că $\mathbf{B}_2^n$ este închis față de operația $\oplus$ că θ este elementul nul al lui $\mathbf{B}_2^n$ și că fiecare element $a \in \mathbf{B}_2^n$ este propriul său invers, $a = a^{-1}$.

Fie $a, b, c \in \mathbf{B}_2^n$. Dacă a_i, b_i, c_i sînt componentele de ordin i ale lui a, b și respectiv c, atunci din relația

$$(a_i \oplus b_i) \oplus c_i = a_i \oplus (b_i \oplus c_i) = a_i b_i c_i + a_i \bar{b}_i \bar{c}_i + \\ + \bar{a}_i b_i \bar{c}_i + \bar{a}_i \bar{b}_i c_i, \ (\forall)\ i \in \{1, 2, \ldots, n\},$$

rezultă $(a \oplus b) \oplus c = a \oplus (b \oplus c)$, ceea ce demonstrează că operația $\oplus$ este și asociativă și deci că $\mathbf{B}_2^n$ este un grup cu operația $\oplus$.

Dacă notăm cu $i = 0, 1, 2, \ldots, 2^n - 1$ liniile codului $\mathbf{B}_2^n$, se poate arăta că există o asemenea ordonare a acestor linii, astfel încît să existe relația

$$\mathbf{a}_{2^s+t} = \mathbf{a}_{2^s} \oplus \mathbf{a}_t$$

pentru orice $0 \leqslant s \leqslant n-1$, $0 \leqslant t \leqslant 2^s$. O astfel de ordine a elementelor codului $\mathbf{B}_2^n$ va fi numită *ordine naturală*.

Pentru obținerea acestei ordini este suficient să asociem coloanei j ponderea 2^{n-j} și să ordonăm liniile codului $\mathbf{B}_2^n$ după mărimea numerelor zecimale asociate, date de expresia $z_i = \sum_{j=1}^{n} a_{ij} 2^{n-j}$.

Într-adevăr, $z_0 = 0$ corespunde vectorului nul $\mathbf{a}_0 = (0, 0, \ldots, 0)$.

Apoi z_{2^s} corespunde vectorului $\mathbf{a}_{2^s} = (a_{2^s 1}, a_{2^s 2}, \ldots, a_{2^s n})$ în care : $a_{2^s j} = 0$, $j \neq n - s$, $a_{2^s n-s} = 1$.

Pe de altă parte vectorul $\mathbf{a}_{2^s+t}$ diferă de vectorul $\mathbf{a}_t$, $t < 2^s$, numai prin componenta $j = n - s$, care este egală cu 1 în $\mathbf{a}_{2^s+t}$ și 0 în $\mathbf{a}_t$.

4.2.4. *Spațiul vectorial* $\mathbf{B}_2^n$

După cum s-a văzut, mulțimea vectorilor binari n-dimensionali formează grup față de operația $\oplus$. În afară de operația grupală vom introduce înmulțirea cu scalari, dată de relația $\lambda \mathbf{a}_i = (\lambda a_{i1}, \lambda a_{i2}, \ldots, \lambda a_{in})$, unde λ ia valorile 0 sau 1. Rezultă deci că grupul $\mathbf{B}_2^n$ poate fi organizat ca spațiu vectorial cu operația $\oplus$ și înmulțirea cu scalari. Mai mult, orice subgrup al grupului $\mathbf{B}_2^n$ este un spațiu vectorial față de operația $\oplus$ și înmulțirea cu scalari, deci orice cod grupal are această proprietate.

Vom putea deci introduce noțiunile de dependență și independență liniară între liniile codurilor grupale.

Definiție. Liniile $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ ale unui cod grupal $\mathbf{K}$ se numesc liniar independente, dacă egalitatea

$$\lambda_1 \mathbf{a}_1 \oplus \lambda_2 \mathbf{a}_2 \oplus \ldots \oplus \lambda_m \mathbf{a}_m = \bigoplus_{i=1}^{m} \lambda_i \mathbf{a}_i = \theta,$$

unde $\lambda_i \mathbf{a}_i = \mathbf{a}_i$ pentru $\lambda_i = 1$ și $\lambda_i \mathbf{a}_i = 0$ pentru $\lambda_i = 0$ are loc numai dacă toți $\lambda_i = 0$ și sînt liniar dependente în caz contrar. Prin θ am notat vectorul cu toate componentele nule.

Ținînd seama de relația de independență, se deduce că un spațiu vectorial cu 2^m vectori are un sistem de generatori format din m vectori liniar independenți, m fiind dimensiunea spațiului. Într-adevăr, fie $\mathbf{a} = \bigoplus_{i=1}^{n} \lambda_i \mathbf{a}_i$ și $\mathbf{a}' = \bigoplus_{i=1}^{n} \lambda'_i \mathbf{a}_i$ două combinații liniare ale vectorilor $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ liniar independenți. Dacă $(\lambda_1, \lambda_2, \ldots, \lambda_m) \neq (\lambda'_1, \lambda'_2, \ldots, \lambda'_m)$, atunci $\mathbf{a} \neq \mathbf{a}'$. Altfel din $\mathbf{a} = \mathbf{a}'$ se deduce

$$\mathbf{a} \oplus \mathbf{a}' = \bigoplus_{i=1}^{n} (\lambda_i \oplus \lambda'_i)\, \mathbf{a}_i = \theta,$$

unde cel puțin un scalar $\lambda_i \oplus \lambda'_i$ este diferit de zero, ceea ce ar conduce la concluzia că vectorii $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ ar fi liniar dependenți, contrar presupunerii.

Fie **P** și **Q** subspații vectoriale de dimensiune p și respectiv q. Prin definiție se numește *intersecție* a subspațiilor **P** și **Q** mulțimea vectorilor care sînt incluși și în subspațiul **P** și în subspațiul **Q**. Se demonstrează că intersecția a două subspații este un subspațiu.

Se numește *reuniune* a două subspații **P** și **Q** subspațiul generat de reuniunea bazelor celor două subspații. Dacă s este dimensiunea reuniunii spațiilor **P** și **Q** iar t dimensiunea intersecției lor, atunci există relația $p + q = s + t$, unde p și q sînt dimensiunile lui **P** și respectiv **Q**.

4.2.5. *Coduri duale*

Să considerăm două coduri **K** și **K*** de dimensiune p și respectiv q și fie

$\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, $i = 0, 1, 2, \ldots, 2^{p-1}$, liniile codului **K** și $\mathbf{c}_k = (c_{k1}, c_{k2}, \ldots, c_{kn})$, $k = 0,1,2, \ldots, 2^{q-1}$, liniile codului **K***.

Definiție. Codurile K și K* se numesc *duale*, dacă între liniile lor există relațiile $\sum_{j=1}^{n} a_{ij}c_{kj} = 0$, $i = 0,1, 2, \ldots, 2^{p-1}$, $k = 0,1,2, \ldots, 2^{q-1}$, denumite relații de dualitate , și dacă sînt maximale, adică dacă nu există un cod strict mai mare decît **K** sau un cod strict mai mare decît **K***, în sensul relației de incluziune, pentru care să se verifice relațiile de dualitate.

Din relațiile de dualitate rezultă *ortogonalitatea codurilor duale.*

Exemplu. Codurile **K** și **K*** date de matricele din tabelele 4.2 și 4.3 sînt *coduri grupale duale*, după cum se poate verifica.

Întrucît intersecția celor două coduri este diferită de vectorul nul, fiind constituită din spațiul cu o dimensiune din tabelul 4.4, iar cele două coduri au dimensiuni egale cu trei, rezultă că reuniunea lor va fi de dimensiunea cinci.

După cum se va vedea, codurile duale joacă un rol important în sinteza multipolilor, una din etapele de sinteză a automatelor finite.

Tabelul 4.2

	a_{i_1}	a_{i_2}	a_{i_3}	a_{i_4}	a_{i_5}	a_{i_6}
a_0	0	0	0	0	0	0
a_1	1	0	1	0	0	1
a_2	0	1	0	1	0	1
a_3	1	1	1	1	0	0
a_4	1	0	0	1	1	1
a_5	0	0	1	1	1	0
a_6	1	1	0	0	1	0
a_7	0	1	1	0	1	1

Tabelul 4.3

	c_{k1}	c_{k2}	c_{k3}	c_{k4}	c_{k5}	c_{k6}
c_0	0	0	0	0	0	0
c_1	1	1	0	0	0	1
c_2	0	0	1	1	0	1
c_3	1	1	1	1	0	0
c_4	1	0	1	0	1	1
c_5	0	1	1	0	1	0
c_6	1	0	0	1	1	0
c_7	0	1	0	1	1	1

Tabelul 4.4

	b_{i_1}	b_{i_2}	b_{i_3}	b_{i_4}	b_{i_5}	b_{i_6}
b_0	0	0	0	0	0	0
b_1	1	1	1	1	0	0

Codurile din tabelele 4.2 și 4.3 pot fi utilizate pentru codificarea numerelor în sistemul de numerație octal. Avînd distanța Hamming egală cu trei, vor corecta o eroare și vor detecta două erori.

4.2.6. *Laticea codurilor bivalente*

Între două linii ale codului $\mathbf{B}_2^n$ vom introduce operațiile de adunare $+$ și înmulțire $\cdot$ definite astfel :

$$\mathbf{a}_i + \mathbf{a}_k = (a_{i1} + a_{k1},\ a_{i2} + a_{k2}, \ldots, a_{in} + a_{kn}),$$

$$\mathbf{a}_i\mathbf{a}_k = (a_{i1}\, a_{k1}, a_{i2}\, a_{k2}, \ldots, a_{in}\, a_{kn}),$$

unde $a_{ij} + a_{kj}$ este suma iar $a_{ij}\, a_{kj}$ este produsul variabilelor booleene a_{ij}, a_{kj}, $(\forall) j = 1,2, \ldots, n$.

Vom introduce de asemenea o relație de divizibilitate.

Definiția 1. Linia $\mathbf{a}$ se numește un *divizor* al liniei $\mathbf{a}_k$ și se scrie $\mathbf{a}_i \leqslant \mathbf{a}_k$, dacă $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_k) = \mathbf{a}_k$, adică dacă $a_{ij} + (a_{ij} \oplus a_{kj}) = a_{kj}$, $j = 1, 2, \ldots, n$.

Lema 1. $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_k) = \mathbf{a}_k \Leftrightarrow \mathbf{a}_i\, \overline{\mathbf{a}}_k = \mathbf{0}$, $\mathbf{0}$ *fiind vectorul cu toate componentele nule*, iar $\overline{\mathbf{a}}_k = (\bar{a}_{k1}, \bar{a}_{k2}, \ldots, \bar{a}_{kn})$.

Demonstrație. Fie x și y două variabile booleene. Atunci

$$x + (x \oplus y) = y \Leftrightarrow x + \bar{x}y + x\bar{y} = y \Leftrightarrow x + \bar{x}y = y \Leftrightarrow \bar{x}(x + \bar{y})y + (x + \bar{x}y)\bar{y} = 0 \Leftrightarrow x\bar{y} = 0.$$

Dar atunci relațiile precedente vor fi adevărate pentru toate componentele vectorilor $\mathbf{a}_i$ și $\mathbf{a}_k$, ceea ce demonstrează lema.

Lema 2. *Relația de divizibilitate este o relație de ordine parțială.*

Demonstrație. Într-adevăr :

1) $\mathbf{a}_i \leqslant \mathbf{a}_i$, deoarece $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_i) = \mathbf{a}_i$.

2) Din $\mathbf{a}_i \leqslant \mathbf{a}_k$ și $\mathbf{a}_k \leqslant \mathbf{a}_i$ se deduce $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_k) = \mathbf{a}_k$, $\mathbf{a}_k + (\mathbf{a}_k \oplus \mathbf{a}_i) = \mathbf{a}_i$ și deoarece $\mathbf{a}_i \oplus \mathbf{a}_k = \mathbf{a}_k \oplus \mathbf{a}_i$, se deduce $\mathbf{a}_i \oplus \mathbf{a}_k = \mathbf{0}$, adică $\mathbf{a}_i = \mathbf{a}_k$.

3) Din $\mathbf{a}_i \leqslant \mathbf{a}_j$ și $\mathbf{a}_j \leqslant \mathbf{a}_k$ se obțin relațiile $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_j) = \mathbf{a}_j$, $\mathbf{a}_j + (\mathbf{a}_j \oplus \mathbf{a}_k) = \mathbf{a}_k$ care sînt echivalente cu $\mathbf{a}_i\, \overline{\mathbf{a}}_j = \mathbf{0}$, $\mathbf{a}_j\, \overline{\mathbf{a}}_k = \mathbf{0}$ și respectiv cu ecuația $\mathbf{a}_i\, \overline{\mathbf{a}}_j + \mathbf{a}_j\, \overline{\mathbf{a}}_k = \mathbf{0}$ care admite soluția $\mathbf{a}_j = (\mathbf{a}_i, \mathbf{a}_k)$ cu condiția $\mathbf{a}_i\, \overline{\mathbf{a}}_k = \mathbf{0}$ sau,

ceea ce este același lucru, cu condiția $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_k) = \mathbf{a}_k$, de unde rezultă $\mathbf{a}_i \leqslant \mathbf{a}_k$.

Lema 3. Suma $\mathbf{a}_i + \mathbf{a}_k$ este cel mai mic multiplu comun iar produsul $\mathbf{a}_i\mathbf{a}_k$ cel mai mare divizor comun al celor două linii.

Demonstrație. Este suficient să se facă demonstrația pe componente.

Fie x o componentă oarecare a lui $\mathbf{a}_i$ și y componenta corespunzătoare a lui $\mathbf{a}_k$. Se verifică imediat că $xy \leqslant x \leqslant x + y$ și $xy \leqslant y \leqslant x + y$. Rămîne să arătăm că xy este cel mai mare element care satisface aceste relații și $x + y$ cel mai mic.

Să arătăm mai întîi că xy este cel mai mare divizor comun al lui x și y. Presupunem că există $z \geqslant y$, astfel încît $xz \leqslant x \leqslant x + y$, $xz \leqslant y \leqslant x + y$ și în plus $z \leqslant x + y$. Atunci se obține succesiv :

$$z \geqslant y \Rightarrow y\bar{z} = 0, \quad xz \leqslant y \Rightarrow xz\bar{y} = 0, \quad z \leqslant x + y \Rightarrow z\bar{x}\bar{y} = 0,$$

de unde $y\bar{z} + xz\bar{y} + z\bar{x}\bar{y} = y\bar{z} + z\bar{y} = 0 \Rightarrow z = y$, ceea ce demonstrează prima parte a lemei.

Să arătăm acum că $z + y$ este cel mai mic multiplu comun al lui x și y. Presupunem că există $z \leqslant y$ astfel încît $xy \leqslant x \leqslant x+z$, $xy \leqslant y \leqslant y + z$ și în plus $z \geqslant xy$. Se obține succesiv : $z \leqslant y \Rightarrow z\bar{y} = 0$, $y \leqslant x + z \Rightarrow y\bar{x}\bar{z} = 0$, $z \geqslant xy \Rightarrow \bar{z}xy = 0$, de unde $z\bar{y} + y\bar{x}\bar{z} + \bar{z}xy = z\bar{y} + y\bar{z} \Rightarrow z = y$, ceea ce trebuia demonstrat.

Teoremă. *Codul binar* B_2^n *poate fi organizat ca latice cu ajutorul relației de divizibilitate.*

Demonstrație. Teorema rezultă ca o consecință din lemele 2 și 3.

4.2.7. *Grupul divizorilor. Hiperfețe*

Fie $\mathbf{a}$ o linie arbitrară din $\mathbf{B}_2^n$ și $\mathbf{G}(\mathbf{a})$ mulțimea divizorilor lui $\mathbf{a}$.

Teorema 1. *Dacă* $\mathbf{a}$ *este un element din* $\mathbf{B}_2^n$ *și* $\mathbf{G}(\mathbf{a})$ *mulțimea divizorilor lui* $\mathbf{a}$ *incluși în* $\mathbf{B}_2^n$, *atunci* $\mathbf{G}(\mathbf{a})$ *formează grup cu operația* $\oplus$.

Demonstrație. $\theta \in G(a)$, întrucît $\theta + (a \oplus \theta) = a$ pentru orice $a \in B_2^n$. Apoi odată cu b și c, este divizor al lui a și $b \oplus c$. Într-adevăr, $b \leqslant a \Rightarrow b\bar{a} = \theta$ și $c \leqslant a \Rightarrow c\bar{a} = \theta$, adică $(b \oplus c)\bar{a} = b\bar{c}\,\bar{a} + \bar{b}\,c\,\bar{a} = \theta \Rightarrow b \oplus c \leqslant a$. De asemenea $a \oplus a = \theta \Rightarrow a^{-1} = a$, ceea ce demonstrează teorema, ținînd seama că B_2^n este grup cu operația $\oplus$.

Exemplu. $G(1110) = \{1110, 1100, 1010, 0110, 1000, 0100, 0010, 0000\}$. Acest grup este generat de oricare dintre trei vectori nenuli ai săi și se poate obține din B_2^3 prin completarea la dreapta cu o coloană formată din elemente egale cu 0.

Definiție. Numim *hiperplan* în B_2^n subgrupul $G(a)$ al divizorilor lui a, pentru orice $a \in B_2^n$. Evident vectorul nul și grupul B_2^n formează hiperplane în B_2^n.

Fie $G(a)$ și $G(b)$ două subgrupuri ale lui B_2^n, astfel încît $a \leqslant b$. Se poate demonstra că $G(a) \subset G(b)$.

Definiție. Un *subgrup* $G(a)$ al lui $G(b)$ este *maximal* în $G(b)$ dacă $G(a) \subset G(b)$ și dacă nu există o linie c, $a \leqslant c \leqslant b$, astfel încît $G(a) \subset G(c) \subset G(b)$.

Există, în general, mai multe subgrupuri maximale într-un subgrup dat.

Exemplu. Subgrupurile maximale ale grupului $G(1110)$ sînt subgrupurile $G(1100)$, $G(1010)$ și $G(0110)$.

Definiție. Numim *hiperfață* în B_2^n un hiperplan $G(a)$ al lui B_2^n sau mulțimea de elemente din $G(a)$ rămase după eliminarea din $G(a)$ a elementelor conținute într-unul sau mai multe subgrupuri maximale ale lui $G(a)$.

Exemplul 1. Hiperfețele obținute din grupul $G(1110)$ prin eliminarea elementelor subgrupului $G(1100)$ și respectiv subgrupurilor $G(1100)$ și $G(1010)$ sînt $H_1 = \{1110, 1010, 0110, 0010\}$ și respectiv $H_2 = \{1110, 0110\}$.

Fiecare element a din B_2^n este o hiperfață deoarece se obține din $G(a)$ prin eliminarea din acesta a elementelor cuprinse în toate subgrupelor sale maximale. De exemplu, dacă în grupul $G(1110)$ se elimină elementele subgrupurilor $G(1100)$, $G(1010)$ și $G(0110)$, se obține elementul $\{1110\}$.

Se poate demonstra ușor că numărul de subgrupuri maximale într-un grup $G(a)$ *este egal cu* $n-1$, *unde* n *este numărul de* 1 *în linia* a.

Teorema 2. *Fie* H *o hiperfață a lui* B_2^n *și* $G(a)$ *cel mai mic grup care include pe* H, $a \in H$. *Atunci există un element* $b \in H$, $b \leqslant a$, *astfel încît orice element* $c \in H$ *se divide prin* b, $b \leqslant c$.

Demonstrație. Într-adevăr, dacă $c \in H$, atunci orice element d, astfel încît $c \leqslant d \leqslant a$, este cuprins în H. Apoi dacă $c \in H$ și $d \in H$, cu c și d necomparabile, atunci $cd \in H$, deoarece cd va fi eliminat din $G(a)$, fie odată cu c, fie odată cu d, fie odată cu amîndouă aceste elemente. Dacă b este cel mai mic element din H, atunci luăm $b = cd$ și teorema este demonstrată. Dacă nu, există un alt element $e \in H$ care nu se divide cu cd. Dar atunci conform celor arătate $cde \in H$ și toți multiplii lui cde sînt cuprinși în H. Mulțimea elementelor lui H fiind finită, prin procedeul menționat se va ajunge la un element b care se bucură de proprietatea $b \in H$ și toți multiplii lui b și numai aceștia sînt incluși în H, ceea ce demonstrează teorema.

Exemplul 2. Hiperfața H_1 din exemplul 1 conține toate elementele x din B_2^4 care se bucură de proprietatea $(0010) \leqslant x \leqslant (1110)$ după cum se poate verifica ușor.

Ținînd seama de teorema 2 rezultă că orice hiperfață din B_2^n este complet determinată de primul element b și de ultimul element a. De aceea convenim să notăm hiperfețele din B_2^n sub forma (b, a), știind că odată cu b în hiperfața $H = (b, a)$ sînt cuprinși toți multiplii lui b care sînt divizorii lui a; $H = \{x | b \leqslant x \leqslant a\}$.

Definiție. Spunem că o *hiperfață* (a,b) este *inclusă* într-o *hiperfață* (c, d) și scriem $(a, b) \subset (c, d)$ dacă $a \geqslant c$, $b \leqslant d$.

Dacă $F = \{a_1, a_2, \ldots, a_n\}$ este o mulțime de elemente din B_2^n, o hiperfață $(c, d) \subset F$ este maximală în F dacă nu se include în nici o hiperfață a lui F.

4.3. Coduri k-valente

4.3.1. *Codificarea mulțimilor de elemente prin coduri K-valente*

Să considerăm mulțimea vectorilor cu n componente $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, $i = 1, 2, \ldots, n$, în care a_{ij} pentru orice i și j va putea lua valorile $0, 1, 2, \ldots, k_j - 1$. În general $k_j \neq k_i$ pentru $j \neq i$. Vom nota această mulțime cu $\mathbf{B}_k^n$. Se demonstrează cu ușurință că numărul de vectori ai mulțimii $\mathbf{B}_k^n$ este dat de expresia $\mathbf{N} = \prod_{j=1}^{n} k_j$.

Fie $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$ o mulțime de obiecte.

Definiție. Se spune că mulțimii $\mathbf{A}$ i s-a asociat un *cod k-valent* dacă s-a stabilit o corespondență biunivocă între mulțimea $\mathbf{A}$ și o parte $\mathbf{K} \subset \mathbf{B}_k^n$. Submulțimea $\mathbf{K} \subset \mathbf{B}_k^n$ se numește *codul k-valent* al mulțimii $\mathbf{A}$.

Evident pentru a se putea codifica mulțimea $\mathbf{A}$ printr-un cod k-valent este necesar și suficient ca $\prod_{j=1}^{n} k_j \geqslant m$.

Ca și în cazul binar, codul $\mathbf{K}$ poate fi reprezentat printr-o matrice $\mathbf{K} = (a_{ij})$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$.

Exemplu. Spre exemplificare luăm scrierea în baza 4 a cifrelor de la 0 la 9. Se obține o codificare a cifrelor de la 0 la 9 într-un cod k-valent în care $k_1 = 3$, $k_2 = 4$, $k_1 k_2 > 10$, prezentat în tabelul 4.5.

Tabelul 4.5

i	a_{i1}	a_{i2}	i	a_{i1}	a_{i2}
0	0	0	5	1	1
1	0	1	6	1	2
2	0	2	7	1	3
3	0	3	8	2	0
4	1	0	9	2	1

4.3.2. *Coduri k-valente ponderate*

Ca și în cazul binar se pot stabili anumite reguli de codificare, printre care foarte utilizată este metoda ponderării.

Fie $\mathbf{K}$ un cod k-valent, de n componente. Să asociem componentei j ponderea Q_j și liniei i numărul p_i dat de expresia

$$p_i = \sum_{j=1}^{n} a_{ij} Q_j, \qquad i = 1, 2, \ldots, m.$$

Un cod care verifică relația precedentă se numește *cod ponderat*. Mulțimea $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$ este *mulțimea codificată*, iar matricea $\mathbf{K} = (a_{ij})$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ este *codul*.

Exemplu. Codul din tabelul 4.5 este un cod ponderat în care $Q_1 = 4$, $Q_2 = 1$, după cum se verifică ușor.

Un caz particular important de realizare a codurilor k-valente ponderate este cazul în care ponderile Q_j sînt date de expresiile

$$Q_n = 1,$$

$$Q_j = k_{j+1} k_{j+2} \ldots k_n \qquad j = 1, 2, \ldots, n-1.$$

În acest caz se încadrează și codul din exemplul de mai sus, în care $Q_2 = 1$, $Q_1 = k_2 = 4$.

4.3.3. *Coduri k-valente pseudogrupale*

Dacă a și b sînt două variabile k-valente, între acestea vom introduce operația $\oplus$ dată de tabelul 4.6, care generalizează operația notată cu același semn, definită pentru cazul binar.

Observație. $a \oplus b = \max(a, b) - \min(a, b) = |a - b|$.

Definiția 1. Ținînd seama de observația precedentă, operația $\oplus$ va fi denumită *operația de complementare*.

Fie $\mathbf{K}$ un cod k-valent cu matricea de cod $\mathbf{K} = (a_{ij})$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. Între liniile codului $\mathbf{K}$ vom

introduce o operație de complementare, pe care o notăm tot cu semnul $\oplus$, definită astfel :

$\mathbf{a}_i \oplus \mathbf{a}_l = (a_{i1} \oplus a_{l1},\ a_{i2} \oplus a_{l2},\ \ldots,\ a_{in} \oplus a_{ln})$, unde $a_{ij} \oplus a_{lj}$, este operația de complementare pentru variabilele k_j-valente a_{ij} și a_{lj}.

Tabelul 4.6

+	0	1	2	...	$k-2$	$k-1$
0	0	1	2	...	$k-2$	$k-1$
1	1	0	1	...	$k-3$	$k-2$
2	2	1	0	...	$k-4$	$k-3$
$k-2$	$k-2$	$k-3$	$k-4$	...	0	1
$k-1$	$k-1$	$k-2$	$k-3$	...	1	0

Definiția 2. O mulțime de elemente $\mathbf{G}$ se numește *pseudogrup* dacă verifică sistemul de axiome :

1. $(a \in \mathbf{G}, b \in \mathbf{G})\ \ a \oplus b \in \mathbf{G}$.

2. Există un element θ astfel încît

$$\theta \oplus a = a \oplus \theta = a,\ \ (\forall)\ a \in \mathbf{G}.$$

3. $(\forall)\ a \in \mathbf{G}$ există $a^{-1} \in \mathbf{G}$ astfel încît

$$a \oplus a^{-1} = a^{-1} \oplus a = \theta.$$

Dacă, în plus, $\mathbf{G}$ verifică axioma :

4. $a \in \mathbf{G}, b \in \mathbf{G},\ a \oplus b = b \oplus a$, atunci pseudogrupul este comutativ.

Operația de pseudogrup am notat-o cu semnul $\oplus$, deși nu este neapărat operația dată de tabelul 4.6.

Se observă că pseudogrupul este un grup neasociativ.

Exemplu. Vom da ca exemplu de pseudogrup mulțimea vectorilor k-valenți cu n componente, din $\mathbf{B}_k^n$. Ca operație grupală folosim operația de complementare.

Se poate verifica succesiv că $\mathbf{B}_k^n$ verifică axiomele 1—4. Într-adevăr :

1. Fie $\mathbf{a}_i \in \mathbf{B}_k^n$ și $\mathbf{a}_l \in \mathbf{B}_k^n$. Atunci, ținînd seama de definiția operației de complementare, se obține $a_{ij} \oplus a_{lj} = \max(a_{ij}, a_{lj}) - \min(a_{ij}, a_{lj}) \leqslant k_j - 1$ și deci $\mathbf{a}_i \oplus \mathbf{a}_l$ este un vector k-valent, ceea ce verifică axioma 1.

2. Dacă notăm cu $\mathbf{0}$ vectorul cu toate componentele nule, atunci din $a_{ik} \oplus \mathbf{0} = a_{ik}$ pentru orice k și i rezultă că orice vector $\mathbf{a}_i \in \mathbf{B}_k^n$ verifică axioma 2.

3. Fie $\mathbf{a}_i \in \mathbf{B}_k^n$. Pentru orice $k = 1, 2, \ldots, n$ avem $a_{ik} \oplus a_{ik} = \max(a_{ik}, a_{ik}) - \min(a_{ik}, a_{ik}) = \mathbf{0}$, adică pentru orice $\mathbf{a}_i \in \mathbf{B}_k^n$ există $\mathbf{a}_i^{-1} = \mathbf{a}_i$, ceea ce verifică axioma 3.

4. Axioma 4 rezultă din simetria operației de complementare.

Teoremă. *Mulțimea vectorilor k-valenți, cu n componente, $\mathbf{B}_k^n$ formează pseudogrup, cu operația $\oplus$ de complementare.*

Se observă că în pseudogrupuri ecuația

$$a \oplus x = b, \qquad a, b \in \mathbf{G},$$

nu are soluție întotdeauna, după cum se poate verifica în cazul pseudogrupului $\mathbf{B}_k^n$, cu toate că există invers bilateral. Această proprietate deosebește esențial pseudogrupurile de grupuri. Pe de altă parte, utilizarea pseudogrupului $\mathbf{B}_k^n$ în studiul funcțiilor de variabile polivalente justifică studiul aprofundat al pseudogrupurilor. Singura dificultate mai importantă constă în faptul că în pseudogrupuri toate formulele trebuie scrise cu paranteze, din cauza neîndeplinirii relației de asociativitate.

4.3.4. *Laticea codurilor k-valente*

În mulțimea vectorilor k-valenți, $\mathbf{B}_k^n$, vom introduce o relație de divizibilitate.

Definiție. Fie $\mathbf{a}_i$ și $\mathbf{a}_k$ doi vectori k-valenți. Între acești vectori există relația de divizibilitate, notată $\mathbf{a}_i \leqslant \mathbf{a}_k$, dacă $\mathbf{a}_i + (\mathbf{a}_i \oplus \mathbf{a}_k) = \mathbf{a}_k$, unde $+$ este suma obișnuită vectorială.

Lema 1. *Condiția necesară și suficientă pentru ca între vectorii* $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ *și* $\mathbf{a}_k = (a_{k1}, a_{k2}, \ldots, a_{kn})$ *să existe relația de divizibilitate este dată de condițiile* $a_{ij} \leqslant a_{kj}$, $j = 1, 2, \ldots, n$.

Demonstrație. Din $a_{ij} \leqslant a_{kj}$ rezultă succesiv :

$$a_{ij} \oplus a_{kj} = a_{kj} - a_{ij},\ a_{ij} + (a_{ij} \oplus a_{kj}) = a_{kj}, (\forall) j = 1, 2, \ldots, n.$$

Reciproc, din $a_{ij} + (a_{ij} \oplus a_{kj}) = a_{kj}$ rezultă succesiv $a_{ij} + \max(a_{ij}, a_{kj}) - \min(a_{ij}, a_{kj}) = a_{kj}$, $a_{kj} - a_{ij} = \max(a_{ij}, a_{ikj}) - \min(a_{ij}, a_{kj}) \geqslant 0$, ceea ce demonstrează lema.

Să notăm prin $d_j = \min(a_{ij}, a_{kj})$ și prin $s_j = \max(a_{ij}, a_{kj})$, $j = 1, 2, \ldots, n$.

Lema 2. *Vectorul* $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ *este cel mai mare divizor comun, iar vectorul* $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ *cel mai mic multiplu comun al vectorilor* $\mathbf{a}_i$ *și* $\mathbf{a}_k$ *din* $\mathbf{B}_k^n$.

Demonstrație. Presupunem contrariul. Atunci pentru orice pereche de componente a_{ij}, a_{kj} ale vectorilor $\mathbf{a}_i$ și $\mathbf{a}_k$ există două numere x și y astfel încît

$$d_j \leqslant x \leqslant a_{ij} \leqslant y \leqslant s_j \text{ și } d_j \leqslant x \leqslant a_{kj} \leqslant y \leqslant s_j.$$

Dar $(x \leqslant a_{ij},\ x \leqslant a_{kj}) \Rightarrow x \leqslant \min(a_{ij}, a_{kj}) = d_j$ și $(a_{ij} \leqslant y,\ a_{kj} \leqslant y) \Rightarrow s = \max(a_{ij}, a_{kj}) \leqslant y$ contrar presupunerii, ceea ce demonstrează lema.

Teoremă. *Pseudogrupul* $\mathbf{B}_k^n$ *poate fi organizat ca latice cu ajutorul relației de divizibilitate.*

Demonstrație. Ca și în cazul binar teorema rezultă ca o consecință a lemelor 1 și 2.

4.3.5. *Pseudogrupul divizorilor. Hiperplane*

Fie $\mathbf{a}$ un element arbitrar din $\mathbf{B}_k^n$ și $\mathbf{G}(\mathbf{a})$ mulțimea divizorilor lui $\mathbf{a}$.

Teoremă. *Dacă* $\mathbf{a}$ *este un element din* $\mathbf{B}_k^n$ *și* $\mathbf{G}(\mathbf{a})$ *mulțimea divizorilor lui* $\mathbf{a}$ *incluși în* $\mathbf{B}_k^n$, *atunci* $\mathbf{G}(\mathbf{a})$ *formează pseudogrup cu operația* $\oplus$.

Demonstrație. $\mathbf{0} \in \mathbf{G}(\mathbf{a})$ întrucît $\mathbf{0} + (\mathbf{a} \oplus \mathbf{0}) = \mathbf{a}$ pentru orice $\mathbf{a} \in \mathbf{B}_k^n$. Apoi odată cu $\mathbf{b}$ și $\mathbf{c}$ este divizor al lui $\mathbf{a}$ și $\mathbf{b} \oplus \mathbf{c}$. Într-adevăr :

$$(\mathbf{b} \leqslant \mathbf{a}) \Rightarrow b_i \leqslant a_i, \qquad i = 1, 2, \ldots, n,$$

$$(\mathbf{c} \leqslant \mathbf{a}) \Rightarrow c_i \leqslant a_i, \qquad i = 1, 2, \ldots, n.$$

Dar $\mathbf{b} \oplus \mathbf{c} = (|b_1 - c_1|, |b_2 - c_2|, \ldots, |b_n - c_n|)$ și întrucît

$$|b_i - c_i| = \begin{cases} b_i - c_i, \text{ dacă } b_i > c_i, \\ c_i - b_i, \text{ dacă } b_i < c_i, \\ 0, \text{ dacă } b_i = c_i, \end{cases}$$

rezultă succesiv

$$b_i - c_i \leqslant b_i \leqslant a_i,$$

$$c_i - b_i \leqslant c_i \leqslant a_i,$$

$$0 \leqslant a_i,$$

adică $|b_i - c_i| \leqslant a_i$, $i = 1, 2, \ldots, n$, și respectiv $\mathbf{b} \oplus \mathbf{c} \leqslant \mathbf{a}$, ceea ce trebuia demonstrat.

Exemplul 1. Pentru exemplificare vom considera mulțimea divizorilor elementului $\mathbf{a} = (2, 3)$ din codul

$\mathbf{B}_k^2 = \{(2, 3), (2, 2), (2, 1), (2, 0), (1, 3), (1, 2), (1, 1), (1, 0), (0, 3), (0, 2), (0, 1), (0, 0)\}$.

Se verifică ușor că aceștia formează pseudogrup cu operația $\oplus$.

Definiția 1. Spunem că un pseudogrup $\mathbf{G}(\mathbf{a})$ inclus într-un pseudogrup $\mathbf{G}(\mathbf{b})$ *este maximal* în $\mathbf{G}(\mathbf{b})$ dacă nu există un element $\mathbf{c}$, cu proprietatea $\mathbf{a} \leqslant \mathbf{c} \leqslant \mathbf{b}$, astfel încît $\mathbf{G}(\mathbf{a}) \subset \mathbf{G}(\mathbf{c}) \subset \mathbf{G}(\mathbf{b})$.

Exemplul 2. Pseudogrupurile maximale ale pseudogrupului $\mathbf{G}(\mathbf{a})$ cu $\mathbf{a} = (2, 3)$ sînt $\mathbf{G}(\mathbf{b})$ cu $\mathbf{b} = (2, 2)$ și $\mathbf{G}(\mathbf{c})$ cu $\mathbf{c} = (1, 3)$.

Există, în general, mai multe pseudogrupuri maximale într-un pseudogrup dat.

Definiția 2. Numim *hiperplan* în $\mathbf{B}_k^n$ un pseudogrup $\mathbf{G(a)}$ al lui $\mathbf{B}_k^n$ generat de vectorul $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ care are componentele egale fie cu 0, fie cu $k_i - 1$.

Exemplul 3. În pseudogrupul $\mathbf{B}_k^2$, în care $k_1 = 3$, $k_2 = 4$, hiperplanele diferite de hiperplanul nul și hiperplanul $\mathbf{B}_k^2$ sînt $\mathbf{G}(2, 0) = \{(2, 0), (1, 0), (0, 0)\}$, $\mathbf{G}(0, 3) = \{(0, 3), (0, 2), (0, 1), (0, 0)\}$.

4.3.6. *Hiperfețe. Cvasifețe.*

Definiția 1. Numim *hiperfață* în $\mathbf{B}_k^n$ o submulțime $\mathbf{H}$ de elemente din $\mathbf{B}_k^n$ pentru care există un element $\mathbf{a} \in \mathbf{H}$ și un hiperplan $\mathbf{G(a)}$ cu proprietatea că pentru orice $\mathbf{x} \in \mathbf{H}$, $\mathbf{a} \oplus \mathbf{x} \in \mathbf{G(a)}$ și aplicația $f_a : \mathbf{H} \to \mathbf{G(a)}$ definită de $f_a(\mathbf{x}) = \mathbf{a} \oplus \mathbf{x}$ este surjecție.

Exemplu. Mulțimea $\mathbf{H} = \{(1, 3), (1, 2), (1, 1), (1, 0)\}$ este o hiperfață în $\mathbf{B}_k^2$. Într-adevăr, luînd $\mathbf{a} = (1, 0)$, se obține $f_a(\mathbf{H}) = \mathbf{G(b)}$, $\mathbf{b} = (0, 3)$. Prin $f_a(\mathbf{H})$ am notat imaginea lui $\mathbf{H}$ prin f_a.

Caz particular. Orice element $\mathbf{a} \in \mathbf{B}_k^n$ este o hiperfață, întrucît aplicația $f_a : \mathbf{a} \to \mathbf{G(\theta)}$ duce pe $\mathbf{a}$ în pseudogrupul nul. De asemenea sînt hiperfețe pseudogrupul $\mathbf{B}_k^n$ și hiperplanele acestuia.

Definiția 2. Două elemente $\mathbf{a}$ și $\mathbf{b}$ ale pseudogrupului $\mathbf{B}_k^n$ se numesc *cvasivecine*, dacă $\mathbf{a} \oplus \mathbf{b}$ are o componentă și numai una diferită de $\mathbf{\theta}$. Fie $a_i \oplus b_i$ această componentă. Dacă $a_i \oplus b_i = 1$, elementele $\mathbf{a}$ și $\mathbf{b}$ se numesc *vecine*.

Definiția 3. Un pseudogrup $\mathbf{G(a)}$ este *cvasimaximal* în $\mathbf{G(b)}$, dacă $\mathbf{G(a)} \subset \mathbf{G(b)}$ și dacă $\mathbf{a}$ și $\mathbf{b}$ sînt cvasivecine.

Există, în general, mai multe pseudogrupuri cvasimaximale într-un pseudogrup dat.

Exemplu. Pseudogrupurile cvasimaximale în pseudogrupul $\mathbf{G(a)}$, $\mathbf{a} = (2, 3)$, sînt: $\mathbf{G}(2, 2)$, $\mathbf{G}(2, 1)$, $\mathbf{G}(2, 0)$, $\mathbf{G}(1, 3)$, $\mathbf{G}(0, 3)$.

Definiția 4. Numim *cvasifață* în $\mathbf{B}_k^n$ un pseudogrup $\mathbf{G(a)} \subset \mathbf{B}_k^n$ sau mulțimea de elemente rămase în $\mathbf{G(a)}$ după eliminarea din acesta a elementelor cuprinse într-unul sau în mai multe pseudogrupuri cvasimaximale.

Orice hiperfață este în același timp și o cvasifață. Pentru grupul $\mathbf{B}_k^n$ noțiunile de hiperfață și de cvasifață se confundă.

Exemplu. Mulțimea $\mathbf{H} = \{(2, 3), (2, 2), (1, 3), (1, 2)\}$ este cvasifață a pseudogrupului $\mathbf{G(a)}$, $\mathbf{a} = (2, 3)$, obținută prin eliminarea din $\mathbf{G(a)}$ a pseudogrupurilor cvasimaximale $\mathbf{G(b)}$, $\mathbf{b} = (2, 1)$ și $\mathbf{G(c)}$, $\mathbf{c} = (0, 3)$.

Teoremă. *Dacă* $\mathbf{H}$ *este o cvasifață în* $\mathbf{B}_k^n$, *în* $\mathbf{H}$ *există două elemente* $\mathbf{a}$ *și* $\mathbf{b}$, *astfel încît pentru orice* $\mathbf{d} \in \mathbf{H}$, *există relațiile* $\mathbf{a} \leqslant \mathbf{d} \leqslant \mathbf{b}$ *și orice* $\mathbf{d} \in \mathbf{B}_k^n$ *pentru care* $\mathbf{a} \leqslant \mathbf{d} \leqslant \mathbf{b}$ *este cuprins în* $\mathbf{H}$, *adică* $\mathbf{H}$ *este complet determinată de primul său element* $\mathbf{a}$ *și de ultimul element* $\mathbf{b}$.

Demonstrație. Fie $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathbf{B}_k^n$ și $\mathbf{G(c)}$ pseudogrupul din care s-a obținut $\mathbf{H}$. Atunci evident $\mathbf{b} = \mathbf{c}$.

Să considerăm s vectori din $\mathbf{G(c)}$, $\mathbf{r}_i = (r_{i1}, r_{i2} \ldots, r_{in})$, $i = 1, 2, \ldots, s$, cvasivecini cu vectorul $\mathbf{c}$ și să presupunem că din $\mathbf{G(c)}$ au fost eliminate pseudogrupurile cvasimaximale $\mathbf{G}(\mathbf{r}_i)$. Presupunem în plus că vectorii $\mathbf{r}_i$ sînt necomparabili. Altfel, dacă de exemplu $\mathbf{r}_k \leqslant \mathbf{r}_j$, atunci odată cu $\mathbf{G}(\mathbf{r}_j)$ a fost eliminat și pseudogrupul $\mathbf{G}(\mathbf{r}_k)$ și deci vectorul $\mathbf{r}_k$ poate fi lăsat la o parte.

Fie $\mathbf{p}_i$ vectorii dați de expresia: $\mathbf{p}_i = \mathbf{r}_i \oplus \mathbf{c}$, $i = 1, 2, \ldots, s$. Întrucît pseudogrupul $\mathbf{G}(\mathbf{r}_i)$ este cvasimaximal în $\mathbf{G(c)}$, vectorul $\mathbf{p}_i$ are una și numai o componentă diferită de zero. Fie p_{ij_i} componenta diferită de zero în $\mathbf{p}_i$, $i = 1, 2, \ldots, s$. Evident $p_{ij_i} \geqslant 1$, de unde rezultă că $r_{ij_i} + 1 \leqslant r_{ij_i} + p_{ij_i} = c_{ij_i}$.

Să considerăm vectorul $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ în care $a_{j_i} = r_{ij_i} + 1$, $i = 1, 2, \ldots, s$, și $a_k = 0$, $k \neq j_i$, $i = 1, 2, \ldots, s$.

Fie $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ un vector arbitrar din $\mathbf{H}$. Vom arăta că $\mathbf{d} \geqslant \mathbf{a}$, adică $\mathbf{a}$ îndeplinește rolul de prim element al lui $\mathbf{H}$. Într-adevăr, pentru $k \neq j_i$, $i = 1, 2, \ldots, s$, $d_k \geqslant a_k$, întrucît $a_k = 0$. Să arătăm apoi că pentru $k = j_i$, $i = 1, 2, \ldots, s$, $d_k \geqslant a_k$.

Presupunem contrariul. Există atunci cel puțin un $i \leqslant s$, astfel încît $d_k < a_{j_i}$, dar atunci ținînd seama de definiția lui a_{j_i}, se deduce că $d_k \leqslant r_{ij_i}$ și deoarece pentru $l \neq j_i$, $r_{il} = c_l \geqslant d_l$, rezultă că $\mathbf{d} \leqslant \mathbf{r}_i$ și deci $\mathbf{d}$ a fost

eliminat din $\mathbf{G}(\mathbf{c})$ odată cu $\mathbf{G}(\mathbf{r}_i)$, adică $\mathbf{d} \notin \mathbf{H}$ contrar presupunerii.

Să arătăm mai departe că vectorul $\mathbf{a}$ obținut ca mai înainte este inclus în $\mathbf{H}$ și îndeplinește rolul de prim element. Evident $\mathbf{a} \leqslant \mathbf{c} \Rightarrow \mathbf{a} \in \mathbf{G}(\mathbf{c})$. Apoi $a_{j_i} = r_{ij_i} + 1 \Rightarrow \mathbf{a} \leqslant \mathbf{r}_i$, $(\forall) i = 1, 2, \ldots, s$, deci nu a fost eliminat din $\mathbf{G}(\mathbf{c})$ odată cu $\mathbf{G}(\mathbf{r}_j)$.

Deci $\mathbf{a} \in \mathbf{H} = \mathbf{G}(\mathbf{c}) - \bigcup_{i=1}^{s} \mathbf{G}(\mathbf{r}_i)$.

Convenim să notăm cvasifața $\mathbf{H}$ prin perechea $\mathbf{H} = (\mathbf{a}, \mathbf{b})$, unde $\mathbf{a}$ este primul element, iar $\mathbf{b}$ este ultimul element al lui $\mathbf{H}$. În mulțimea cvasifețelor introducem o relație de ordine parțială definită astfel:

Definiția 5. Spunem că o *cvasifață* $(\mathbf{a}, \mathbf{b})$ *este inclusă în cvasifața* $(\mathbf{c}, \mathbf{d})$ și notăm $(\mathbf{a}, \mathbf{b}) \subset (\mathbf{c}, \mathbf{d})$, dacă $\mathbf{a} \geqslant \mathbf{c}$, $\mathbf{b} \leqslant \mathbf{d}$.

Exemplu. În cvasifața $\mathbf{H} = \{(2, 3), (2, 2), (1, 3), (1, 2)\}$ primul element este $\mathbf{a} = (1, 2)$, iar ultimul element este $\mathbf{b} = (2, 3)$. Se verifică imediat că $\mathbf{a}$ divide orice element, în timp ce $\mathbf{b}$ se divide cu orice element din cvasifață.

4.4. Codificarea funcțiilor booleene

4.4.1. *Constituenți ai unității. Implicanți*

Înainte de a trece la codificarea funcțiilor booleene vom da mai întîi definițiile de *constituent al unității*, *implicant* și *implicant prim*.

Definiția 1. Un produs $x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ în care $a_i \in \{0, 1\}$, $(\forall)\, i \in \mathbf{I}_n$, se va numi *constituent al unității*, unde $\mathbf{I}_n = \{1, 2, \ldots, n\}$.

Se poate verifica egalitatea $\sum_{\mathbf{a} \in \mathbf{B}_2^n} x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n} = 1$, unde $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ este vectorul exponenților, ceea ce justifică definiția de constituent al unității.

Fie $f(x_1, x_2, \ldots, x_n)$ o funcție booleană de n variabile.

Definiția . Un produs de k variabile $Y = y^{a_1} y_2^{a_2} \ldots y_k^{a_k}$, $y_i \in \{x_1, x_2, \ldots, x_n\}$, $(\forall) i \in \{1, 2, \ldots, k\}$, este un *implicant al unității*. Dacă $Y \Rightarrow f$, unde $Y \Rightarrow f$, și dacă $Y = 1$ implică $f = 1$, atunci Y se va numi *implicant al funcției* f.

Fie $Y = y_1^{a_1} y_2^{a_2} \ldots y_k^{a_k}$ un implicant al funcției $f(x_1, x_2, \ldots, x_n)$ și Y' produsul obținut din Y dacă se înlătură din el cel puțin o variabilă.

Definiția 3. Implicantul $Y = y_1^{a_1} y_2^{a_2} \ldots y_k^{a_k}$ al funcției f este un *implicant prim* al acestei funcții dacă $Y' \not\Rightarrow f$, adică dacă Y încetează să mai fie *implicant al lui* f de îndată ce înlăturăm din el cel puțin o variabilă.

Implicanții primi sînt minimali în sensul că au un număr minim de variabile, de aceea ei vor juca un rol important în problema simplificării funcțiilor booleene, după cum se va vedea în cap. 6.

Definiția 4. Doi *implicanți* X și Y care diferă prin forma unei singure variabile se numesc *vecini*. Implicanții X și Y sînt atunci de forma $X = U Z^1$, $Y = U Z^0$, unde U este un produs de variabile.

4.4.2. *Codificarea implicanților*

Fie $Y = y_2^{a_1} y_2^{a_2} \ldots y_k^{a_k}$ un implicant al funcției booleene de n variabile $f(x_1, x_2, \ldots, x_n)$ unde $y_1, y_2, \ldots, y_k$ sînt variabile din mulțimea $\{x_1, x_2, \ldots, x_n\}$. Convenim să-i asociem lui Y un vector bivalent de n componente, $\mathbf{V} = (v_1, v_2, \ldots, v_n)$, după următoarea regulă: $(a_i = 1$ și $y_i = x_j) \Rightarrow v_j = 1$ pentru orice $i \in \{1, 2, \ldots, k\}$. Restul de componente ale vectorului $\mathbf{V}$ vor fi egale cu 0. Lui Y îi vom asocia un al doilea vector bivalent de n componente, $\mathbf{U} = (u_1, u_2, \ldots, u_n)$, după regula $x_i \notin \{y_1, y_2, \ldots, y_k\} \Rightarrow u_i = 1$, iar restul de componente ale lui $\mathbf{U}$ vor fi nule.

Definiția 1. Perechea de vectori bivalenți (**U**, **V**) va fi denumită *codul bivalent* al implicantului Y.

Exemplul 1. Implicantul $Y = x_1^1 x_2^0 x_4^1$ al funcției de patru variabile $f(x_1, x_2, x_3, x_4)$ are codul bivalent $\mathbf{W} = (\mathbf{U}, \mathbf{V}) = (0010, 1001)$.

Fie $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ un vector bivalent de n componente.

Definiția 2. Numărul $\mathbf{W}(\mathbf{a}) = \sum_{i \in \mathbf{I}_n} a_{n-i+1} 2^{i-1}$, $\mathbf{I}_n = \{1, 2, \ldots, n\}$, se numește *numărul convențional* al vectorului **a**.

Între vectorii bivalenți și numerele convenționale corespunzătoare se poate stabili o corespondență biunivocă, ceea ce permite scrierea implicanților ca perechi de numere convenționale în loc de perechi de vectori bivalenți. Astfel, implicantul Y poate fi dat prin perechea (**a**,**b**) a numerelor convenționale. Constituenților unității le corespund perechi de numere convenționale de forma $(0, a)$, $a \in \{1,2,\ldots,2^{n-1}\}$, deoarece conțin toate variabilele funcțiilor în care apar.

Definiția 3. Perechea (a, b) de numere convenționale, asociată unui implicant dat Y, se numește *codul convențional* al lui Y.

De regulă se lucrează cu numere convenționale în loc de vectori bivalenți pentru ușurința scrierii. Operațiile cu numere convenționale se înțeleg astfel:

Definiția 4. Fie a, b, c numerele convenționale ale vectorilor bivalenți **u**, **v**, **w** și $\circ$ o lege oarecare de compunere a vectorilor bivalenți. Atunci prin definiție $c = a \circ b \Leftrightarrow \mathbf{w} = \mathbf{u} \circ \mathbf{v}$.

Exemplul 2. Dacă $\mathbf{u} = 1100$, $\mathbf{v} = 1001$, $\mathbf{w} = \mathbf{u} \oplus \mathbf{v} = 0101$, atunci $5 = 12 \oplus 9$, unde 12, 9 și 5 sînt numerele convenționale ale lui **u**, **v** și respectiv **w**.

4.4.3. *Implicanți și hiperfețe*

În acest paragraf vom stabili o corespondență biunivocă între implicanții funcțiilor booleene de n variabile și hiperfețele din spațiul $\mathbf{B}_2^n$, care justifică utilizarea codurilor bivalente în studiul funcțiilor booleene.

Fie X și Y doi implicanți ai unității în $\mathbf{B}_2^n$ astfel încît $Y \Rightarrow X$. Atunci, evident, Y poate fi scris sub forma $Y = XZ$, unde $x_k \in Z \Rightarrow x_k \notin X$ și reciproc $x_k \in X \Rightarrow x_k \notin Z$, iar Z are cel puțin o variabilă, altfel Y nu este un implicant propriu al lui X. Pentru simplificarea scrierii s-au identificat implicanții X și Z cu mulțimile literelor componente.

Fie (**a**, **b**) și (**c**, **d**) codurile bivalente ale lui X și respectiv Y și $\leqslant$ relația de divizibilitate introdusă în 4.2.6.

Lema 1. *Implicația $Y \Rightarrow X$ este echivalentă cu următoarele trei relații* : $\mathbf{c} \leqslant \mathbf{a}$, $\mathbf{a} \leqslant \mathbf{d}$, $\bar{\mathbf{a}}\,\bar{\mathbf{b}} \leqslant \bar{\mathbf{c}}\,\bar{\mathbf{d}}$.

Demonstrație. 1) $(x_k^1 \in X \Rightarrow x_k^1 \in Y) \Rightarrow (a_k = c_k = 0, b_k = d_k = 1)$. 2) $(x_k^0 \in X \Rightarrow x_k^0 \in Y) \Rightarrow (a_k = b_k = c_k = d_k = 0)$. 3) $x_k^1 \cup x_k^0 \in X \Rightarrow a_k = 1$, $b_k = 0$. Din 1), 2) și 3) se deduce imediat $a_k \geqslant c_k$, $b_k \leqslant d_k$, $\bar{a}_k \bar{b}_k \leqslant \bar{c}_k \bar{d}_k$, $(\forall)\, k \in \{1, 2, \ldots, n\} \Rightarrow \mathbf{a} \geqslant \mathbf{c}$, $\mathbf{b} \leqslant \mathbf{d}$, $\bar{\mathbf{a}}\,\bar{\mathbf{b}} \leqslant \bar{\mathbf{c}}\,\bar{\mathbf{d}}$. Reciproc, din $c_k \leqslant a_k$, $b_k \leqslant d_k$, $\bar{a}_k \bar{b}_k \leqslant \bar{c}_k \bar{d}_k$, ținînd seama și de relațiile $a_k b_k = 0$, $c_k d_k = 0$, rezultă : 1) $(a_k = 0,\ b_k = 1) \Rightarrow (c_k = 0,\ d_k = 1) \Rightarrow (x_k^1 \in X \Rightarrow x_k^1 \in Y)$; 2) $(a_k = 0,\ b_k = 0) \Rightarrow (c_k = 0,\ 1 \leqslant \bar{d}_k \Leftrightarrow d_k = 0) \Rightarrow x_k^0 \in X \Rightarrow x_k^0 \in Y)$, $(\forall)\, k \in \{1, 2, \ldots, n\}$, adică $Y \Rightarrow X$.

Lema 2. *Dacă Y este un implicant al unității și* (**a**, **b**) *codul său bivalent, atunci* $\mathbf{ab} = \mathbf{0}$, **0** *fiind vectorul nul.*

Demonstrație. $(a_k = 1 \Rightarrow b_k = 0,\ a_k = 0 \Rightarrow b_k$ arbitrar$) \Rightarrow a_k b_k = 0$, $(\forall)\, k \in \{1, 2, \ldots, n\} \Rightarrow \mathbf{a}\,\mathbf{b} = \mathbf{0}$.

Un implicant al unității poate fi definit, ținînd seama de lema 2, ca o pereche ordonată de numere convenționale (**a**, **b**) între care există relația $\mathbf{ab} = \mathbf{0}$. Ținînd seama de lemele 1 și 2, rezultă că relația de ordine parțială între implicanții unității $X = (\mathbf{a}, \mathbf{b})$ și $Y = (\mathbf{c}, \mathbf{d})$, indusă de implicația $Y \Rightarrow X$, este echivalentă cu relațiile următoare între codurile bivalente : $\mathbf{c} \leqslant \mathbf{a}$, $\mathbf{b} \leqslant \mathbf{d}$, $\bar{\mathbf{a}}\,\bar{\mathbf{b}} \leqslant \bar{\mathbf{c}}\,\bar{\mathbf{d}}$, $\mathbf{ab} = \mathbf{0}$, $\mathbf{cd} = \mathbf{0}$.

Sîntem în măsură să dăm acum o teoremă care stabilește legătura între mulțimea implicanților unității și mulțimea hiperfețelor din spațiul $\mathbf{B}_2^n$. Fiecărui implicant al unității (**a**, **b**) îi vom asocia hiperfața (**b**, $\mathbf{a} + \mathbf{b}$) și, reciproc,

unei hiperfețe $(\mathbf{c}, \mathbf{d})$ îi asociem implicantul unității $(\mathbf{c} \oplus \mathbf{d}, \mathbf{c})$. Se poate verifica ușor că această corespondență este biunivocă.

Teoremă. Dacă $(\mathbf{a}, \mathbf{b})$ *și* $(\mathbf{c}, \mathbf{d})$ *sînt doi implicanți ai unității, atunci* $(\mathbf{c}, \mathbf{d}) \leqslant (\mathbf{a}, \mathbf{b}) \Leftrightarrow (\mathbf{d}, \mathbf{c} + \mathbf{d}) \subset (\mathbf{b}, \mathbf{a} + \mathbf{b})$.

Demonstrație. Obținem succesiv : $(\mathbf{c}, \mathbf{d}) \leqslant (\mathbf{a}, \mathbf{b}) \Rightarrow$ $\Rightarrow (\mathbf{b} \leqslant \mathbf{d}, \bar{\mathbf{a}}\bar{\mathbf{b}} \leqslant \bar{\mathbf{c}}\bar{\mathbf{d}}) \Rightarrow (\mathbf{b} \leqslant \mathbf{d}, \bar{\mathbf{a}}\bar{\mathbf{b}}(\mathbf{c}+\mathbf{d}) = \mathbf{0}) \Rightarrow (\mathbf{b} \leqslant \mathbf{d}, \mathbf{c}+\mathbf{d} \leqslant \mathbf{a}+$ $+ \mathbf{b}) \Rightarrow (\mathbf{d}, \mathbf{c} + \mathbf{d}) \subset (\mathbf{b}, \mathbf{a} + \mathbf{b})$. Apoi dacă $(\mathbf{d}, \mathbf{x})$ și $(\mathbf{b}, \mathbf{y})$ sînt două hiperfețe astfel încît $(\mathbf{d}, \mathbf{x}) \subset (\mathbf{b}, \mathbf{y})$, atunci se obține succesiv : $(\mathbf{b} \leqslant \mathbf{d}, \mathbf{x} \leqslant \mathbf{y}, \mathbf{d} \leqslant \mathbf{x}, \mathbf{b} \leqslant \mathbf{y}) \Leftrightarrow (\mathbf{b}\bar{\mathbf{d}} = \mathbf{0}$, $\mathbf{x}\bar{\mathbf{y}} = \mathbf{0}, \mathbf{d}\bar{\mathbf{x}} = \mathbf{0}, \mathbf{b}\bar{\mathbf{y}} = \mathbf{0}) \Rightarrow \mathbf{d} \oplus \mathbf{x} \leqslant \mathbf{b} \oplus \mathbf{y}$, de unde rezultă că $(\mathbf{b} \oplus \mathbf{y}, \mathbf{b})$ și $(\mathbf{d} \oplus \mathbf{x}, \mathbf{d})$ sînt implicanți ai unității. Luînd, $\mathbf{x} = \mathbf{c} + \mathbf{d}$ și $\mathbf{y} = \mathbf{a} + \mathbf{b}$, rezultă $(\mathbf{d}, \mathbf{c} + \mathbf{d}) \subset (\mathbf{b}, \mathbf{a} + \mathbf{b}) \Rightarrow$ $\leqslant (\mathbf{a}, \mathbf{b})$, adică și implicația inversă, ceea ce demonstrează complet teorema.

4.5. Codificarea funcțiilor de variabile polivalente

4.5.1. *Constituenți ai unității. Implicanți*

Ca și în cazul funcțiilor de variabile bivalente vom da mai întîi noțiunile de constituient al unității, de implicant și implicant prim.

Fie sumele $S(x_i) = \sum_{j \in \mathbf{I}_{k_i - 1}} x_i^j$ și produsul $P = \prod_{i \in \mathbf{I}_n} S(x_i)$, unde $x_1, x_2, \ldots, x_n$ sînt n variabile polivalente, $\mathbf{I}_{k_i - 1} =$ $= \{0, 1, \ldots, k_{i-1}\}$, $\mathbf{I}_n = \{1, 2, \ldots, n\}$. Deoarece $S(x_i) = 1$, $(\forall)\, i \in \mathbf{I}_n$, rezultă $P = 1$. Convenim să asociem produsul P pseudogrupului $\mathbf{B}_k^n$.

Fie apoi sumele parțiale $S_{a_i}(x_i) = \sum_{j \in [p_i, q_i]} x_i^j$, $a_i =$ $= [p_i, q_i]$, $i \in \mathbf{I}_n$, și produsul acestor sume $P_a =$ $= \prod_{i \in \mathbf{I}_n} S_{a_i}(x_i)$. Evident $P_a \Rightarrow P$, $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, ceea ce justifică următoarea definiție.

Definiția 1. Produsul P_a se va numi *constituent al unității* în mulțimea funcțiilor de variabile polivalente.

Produsele de variabile polivalente $x_1^{a_1} x_2^{a_2}, \ldots, x_n^{a_n}$, $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbf{B}_k^n$, $a_i = [p_i, q_i]$, $p_i = q_i$, $i \in \mathbf{I}_n$, sînt cazuri particulare de constituenți ai unității.

Fie $f(x_1, x_2, \ldots, x_n)$ o funcție de variabile polivalente.

Definiția 2. Un produs $Y = \prod_{i \in \mathbf{K}} S_{a_i}(x_i)$, unde $\mathbf{K} \subset \mathbf{I}_n$, este o mulțime de indici, se numește *implicant al unității*. Dacă $Y \Rightarrow f$, atunci Y se va numi *implicant al funcției* f.

Fie Y un implicant al unei funcții f de variabile polivalente și Y' produsul obținut din Y prin înlăturarea cel puțin a unei variabile, ceea ce este același lucru cu înlăturarea din Y a unei sume $S_{a_i}(x_i)$.

Definiția 3. Un implicant Y al unei funcții de variabile polivalente f este un *implicant prim al acestei funcții*, dacă $Y' \not\Rightarrow f$, adică dacă Y încetează să mai fie implicant al lui f de îndată ce se înlătură din el cel puțin o variabilă.

Caracterul minimal al implicantului prim va fi folosit în problema simplificării funcțiilor de variabile polivalente.

Definiția 4. Doi implicanți X și Y sînt *cvasivecini* dacă diferă numai prin forma unei variabile. Atunci ei vor putea fi scriși astfel : $X = \bigcap z^a$, $Y = \bigcap z^b$, unde $\bigcap$ este produs de variabile. Dacă în plus $a \oplus b = 1$, atunci implicanții X și Y se numesc *vecini*

4.5.2. *Codificarea implicanților.*

Fie $Y = \prod_{i \in \mathbf{K}} S_{a_i}(x_i), \mathbf{K} \subset \mathbf{I}_n$, $S_{a_i}(x_i) = \sum_{j \in [p_i, q_i]} x_i^j$, $a_i =$ $= [p_i, q_i]$, un implicant al unității. Convenim să-i asociem lui Y doi vectori k-valenți $\mathbf{u}$ și $\mathbf{v}$, cu n componente fiecare, după regula : $\mathbf{u} = (u_1, u_2, \ldots, u_n)$, $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, unde $u_i = q_i - p_i$, $v_i = p_i$, $(\forall)\, i \in \mathbf{K}$, $u_i = k_i - 1$, $v_i = 0$, în caz contrar.

Consecință. Dacă se notează : $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, $\mathbf{q} = (q_1, q_2, \ldots, q_n)$, p_i, q_i fiind limitele intervalelor $[p_i, q_i]$

pentru $i \in \mathbf{K}$, $p_i = 0$, $q_i = k - 1$ pentru $i \notin \mathbf{K}$, atunci se poate scrie $\mathbf{u} = \mathbf{p} \oplus \mathbf{q}$, $\mathbf{v} = \mathbf{p}$. Deci $u_i + v_i \leqslant k_i - 1$ pentru orice $i \in [1, n]$.

Definiția 1. Perechea $(\mathbf{u}, \mathbf{v})$, $\mathbf{u} = \mathbf{p} \oplus \mathbf{q}$, $\mathbf{v} = \mathbf{p}$, se va numi *codul polivalent al implicantului* $Y = \prod_{i \in \mathbf{K}} S_{a_i}(x_i)$, $\mathbf{K} \subset \mathbf{I}_n$, $S_{a_i}(x_i) = \sum_{j \in [p_i,\ q_i]} x_i^j$, $a_i = [p_i,\ q_i]$, $i \in \mathbf{I}_n$.

Lema 1. a) *Pentru* $\mathbf{u} = \mathbf{0}$, unde $\mathbf{0}$ *este vectorul nul, implicantul unității* Y *se reduce la un produs de variabile.* b) *Lipsa variabilei* x_i *din implicantul* Y *implică* $u_i = k_i - 1$, $v_i = 0$.

Demonstrație. $\mathbf{u} = \mathbf{0} \Rightarrow \mathbf{p} \oplus \mathbf{q} = \mathbf{0} \Rightarrow \mathbf{p} = \mathbf{q}$, de unde rezultă că $S_{a_i}(x_i)$ conține un singur termen pentru orice i, ceea ce demonstrează prima parte a lemei. Pentru demonstrarea părții a doua a lemei observăm că lipsa variabilei x_i dintr-un implicant al unității este echivalentă cu egalitatea $S_{a_i}(x_a) = 1$, de unde $p_i = 0$, $q_i = k_i - 1$, ceea ce implică $u_i = p_i \oplus q_i = k_i - 1, v_i = p_i = 0$.

Lema 1 demonstrează că definiția codului polivalent al unui implicant al unității este o generalizare a noțiunii de cod bivalent. Într-adevăr, lipsa unei variabile x_i dintr-un implicant al unității Y corespunde egalității $u_i = \max x_i$, unde prin $\max x_i$ s-a notat valoarea maximă pe care poate s-o ia variabila x_i, egală cu 1 în cazul bivalent și cu $k_i - 1$ în cazul polivalent. Apoi $u_i < \max x_i$ implică prezența variabilei x_i în implicantul unității Y. Ținînd seama de această observație, cazul bivalent apare ca un caz particular al celui polivalent.

Convenim să asociem variabilelor x_i ponderile w_i, după următoarea regulă : $w_n = 1$, $w_i = k_{i+1} w_{i+1}$.

Definiția 2. Dacă $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ este un vector polivalent, numărul $w(\mathbf{a}) = \sum_{i \in \mathbf{I}_n} a_i w_i$ se numește *numărul convențional al vectorului* $\mathbf{a}$.

Între numerele convenționale și vectorii polivalenți se stabilește, ca și în cazul bivalent, o corespondență biunivocă. Pentru $k_i = 2$, $(\forall)\ i \in \mathbf{I}_n$, definiția 2 se reduce la definiția corespnzătoare din cazul bivalent. Consecința

definiției 2 este că implicanții polivalenți pot fi scriși la fel de bine sub formă de perechi de vectori sau perechi de constituenți ai unității.

Exemplul 1. Implicantul unității $(x_1^1 + x_1^2)(x_2^0 + x_2^1)$ în care fiecare variabilă este trivalentă poate fi reprezentat prin perechea de numere convenționale (4, 3). Într-adevăr, $(\mathbf{p} = (1, 2),\ \mathbf{q} = (0, 1)) \Rightarrow (\mathbf{u} = (1, 1),\ \mathbf{v} = (0, 1)) \Rightarrow (w(\mathbf{u}) = 4, w(\mathbf{v}) = 3)$.

Definiția 3. Perechea (a, b) de numere convenționale asociată unui implicant al unității Y se numește *codul convențional al lui* Y.

Operațiile cu numere convenționale corespund operațiilor cu vectorii asociați.

Definiția 4. Fie a, b, c numerele convenționale asociate vectorilor polivalenți, $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, și $\circ$ o lege oarecare de compunere a vectorilor polivalenți. Prin definiție, $c = a \circ b \Rightarrow \mathbf{w} = \mathbf{u} \circ \mathbf{v}$.

Exemplul 2. $(1, 2) \oplus (0, 1) = (1, 1) \Leftrightarrow 5 \oplus 1 = 4$ în mulțimea vectorilor bidimensionali trivalenți.

4.5.3. *Implicanți și hiperfețe*

Considerăm din nou implicantul unității $P = \prod_{i \in \mathbf{K}} S(p_i, q_i)$; $S(p_i, q_i) = \sum_{j \in [p_i,\ q_i]} x_i^j$, $(\forall)\ i \in \mathbf{K}, \mathbf{K} \subset \mathbf{I}_n$, căruia îi vom asocia hiperfața $\mathbf{H} = (\mathbf{p}, \mathbf{q})$, unde $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, $\mathbf{q} = (q_1, q_2, \ldots, q_n)$, p_i și q_i fiind limitele intervalelor $[p_i, q_i]$ pentru $i \in \mathbf{K}$ și $p_i = 0$, $q_i = k_i - 1$, pentru $i \notin \mathbf{K}$.

Fie X și Y doi implicanți ai unității și $(\mathbf{a}, \mathbf{b})$, respectiv $(\mathbf{c}, \mathbf{d})$ hiperfețele asociate.

Lema 1. *Implicația* $Y \Rightarrow X$ *este echivalentă cu următoarele relații :* $\mathbf{a} \leqslant \mathbf{c}$, $\mathbf{d} \leqslant \mathbf{b}$, *deci cu incluziunea* $(\mathbf{c}, \mathbf{d}) \subset (\mathbf{a}, \mathbf{b})$.

Demonstrație. Fie $X = \prod_{i \in \mathbf{K}_x} S(a_i, b_i)$, $S(a_i, b_i) = \sum_{j \in [a_i,\ b_i]} x_i^j$, $Y = \prod_{i \in \mathbf{K}_Y} S(c_i, d_i)$, $S(c_i, d_i) = \sum_{j \in [c_i, d_i]} x_i^j$, cei doi implicanți, $\mathbf{K}_x$, $\mathbf{K}_y \subset \mathbf{I}_n$.

$(Y = 1) \Leftrightarrow (S(c_i, d_i) = 1, (\forall)\, i \in \mathbf{K}_Y)$ și $(X = 1) \Leftrightarrow$ $\Leftrightarrow (S(a_i, b_i) = 1, (\forall)\, i \in \mathbf{K}_x)$.

Fie $i \in \mathbf{K}_Y$. Pentru orice $m \in [c_i, d_i]$ rezultă $x_i^m \Rightarrow Y$. Apoi din $x_i^m \Rightarrow Y$ și $Y \Rightarrow X$ rezultă $x_i^m \Rightarrow X$. Dacă $i \in \mathbf{K}_X$ atunci $m \in [a_i, b_i]$. Incluziunea fiind adevărată pentru orice $m \in [c_i, d_i]$, este adevărată și pentru c_i și d_i, de unde $a_i \leqslant c_i \leqslant d_i \leqslant b_i$. Dacă $i \notin \mathbf{K}_X$, atunci $[a_i, b_i] = [0, k_i - 1]$ și cu atît mai mult $a_i \leqslant c_i \leqslant d_i \leqslant b_i$.

Fie X și Y doi implicanți ai unității și (a, b), respectiv (c, d) codurile convenționale asociate.

Lema 2. *Implicația $Y \Rightarrow X$ este echivalentă cu următoarele relații : $c \leqslant a$, $b \leqslant d$, $c + d \leqslant a + b$.*

Demonstrație. Considerînd hiperfețele asociate $(b, a+b)$ și $(d, c+d)$ și ținînd seama de lema 1, rezultă $b \leqslant d$, $c+d \leqslant a+b$. Mai departe, din $b \leqslant d$ și $c + d \leqslant a + b$ rezultă $c + d \leqslant a + d \Rightarrow c \leqslant a$, ceea ce demonstrează lema.

REPREZENTĂRILE FUNCȚIILOR BOOLEENE

5.1. Generalități

Studiul funcțiilor booleene se face în multe cazuri pe reprezentările acestora. Există o diversitate mare de reprezentări ale funcțiilor booleene utilizate în mod curent. Aceste reprezentări pot fi grupate în două mari categorii : reprezentările geometrice și reprezentările prin coduri. Reprezentările din prima categorie fiind intuitive, sînt foarte utile pentru studiul funcțiilor booleene cu un număr redus de variabile, fără mijloace electronice de calcul. Pentru funcții cu număr mare de variabile reprezentările geometrice se complică excesiv, devenind practic inutilizabile. Cele mai cunoscute reprezentări geometrice sînt reprezentările pe hipercub, prin grafuri și reprezentările prin diagramele lui Euler, Venn, Veitch și Karnaugh. Întrucît reprezentările geometrice se găsesc în majoritatea lucrărilor de teoria automatelor, vom da numai reprezentarea pe hipercub, care este mai apropiată de a doua categorie de reprezentări, și reprezentările prin grafuri, în cadrul cărora includem schemele secvențiale, schemele cu contacte de relee sau alte elemente tehnice, reprezentabile prin variabile sau funcții booleene. Dealtfel, existența acestor din urmă reprezentări a permis studiul automatelor cu ajutorul funcțiilor booleene.

Reprezentările din a doua categorie, deși sînt mai puțin intuitive, au căpătat o largă răspîndire, întrucît permit studiul funcțiilor cu un număr arbitrar de variabile

și utilizarea calculatoarelor electronice. În această categorie vom prezenta atît reprezentările obișnuite, cît și unele reprezentări propuse de autor, utile în simplificarea funcțiilor booleene sau în sinteza automatelor.

5.2. Reprezentarea pe hipercub

Să considerăm o funcție booleană $f(x_1, x_2, \ldots, x_n)$ scrisă sub forma normală disjunctivă canonică :

$$f(x_1, x_2, \ldots, x_n) = \sum_{\mathbf{a} \in \mathbf{B}_2^n} f(\mathbf{a})\, x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n},$$

$\mathbf{a} = (a_1, a_2, \ldots, a_n)$. Convenim să asociem constituentului unității $P_a(\mathbf{x}) = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ vîrful hipercubului care corespunde vectorului $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ după următoarea regulă :

— Unul din vîrfurile hipercubului se asociază vectorului nul $\boldsymbol{\theta} = (0, 0, \ldots, 0)$.

— Vîrfurile vecine cu $\boldsymbol{\theta}$ se asociază vectorilor $\boldsymbol{\theta}_i$, $i = 1, 2, \ldots, n$, într-o ordine arbitrară. Vectorul $\boldsymbol{\theta}_i$ pentru orice i se bucură de proprietatea că are componenta i egală cu 1 și restul de componente nule. Mulțimea acestor vectori constituie o bază în $\mathbf{B}_2^n$.

— Dacă vectorul $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ se obține din vectorii bazei prin combinația liniară $\mathbf{a} = \sum_{i=1}^{n} p_i \boldsymbol{\theta}_i$, $p_i = 0$ sau 1, i se asociază vîrful $\mathbf{p} = (p_1, p_2, \ldots, p_n)$.

Ținînd seama de definiția vectorilor $\boldsymbol{\theta}_i$, se obține imediat $p_i = a_i$, $(\forall)\ i = 1, 2, \ldots, n$.

Exemplu. Mulțimea constituenților unității pentru funcțiile de trei variabile se reprezintă pe cub ca în fig. 5.1.

O altă reprezentare se poate face ca în fig. 5.2, în care s-a considerat proiecția pe plan a cubului.

Ca regulă generală se procedează astfel :

— Se aranjează vectorii $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ pe $n + 1$ coloane succesive, în coloana i incluzîndu-se toți vectorii

care au $i - 1$ componente egale cu 1 și restul de componente egale cu 0.

— Se unesc prin arce vectorii vecini.

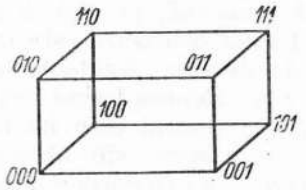

Fig. 5.1

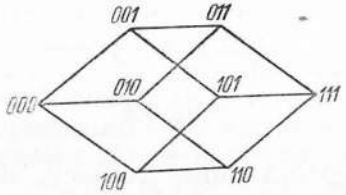

Fig. 5.3

Reprezentarea unei funcții booleene pe hipercub se face prin marcarea cu cercuri pline a vîrfurilor corespunzătoare constituenților unității care apar în forma normală disjunctivă canonică a funcției.

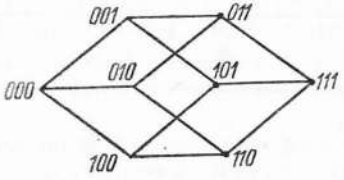

Fig. 5.2

Exemplu. Funcția $f(x_1, x_2, x_3) = x_1^0 x_2^1 x_3^1 + x_1^1 x_2^0 x_3^1 + x_1^1 x_2^1 x_3^0 + x_1^1 x_2^1 x_3^1$ este reprezentată pe hipercub în fig. 5.3 prin nodurile marcate.

5.3. Reprezentarea prin scheme cu contacte de relee

După cum se știe, unui contact de releu x i se poate asocia o variabilă booleană, pe care o notăm tot prin x, care ia valoarea 1 cînd contactul este închis și valoarea 0 cînd contactul este deschis. Evident, se poate face convenția inversă că x ia valoarea 1 cînd contactul este deschis și valoarea 0 cînd contactul este închis. Contactele din prima categorie se numesc de obicei *contacte normal închise*, iar contactele din categoria a doua contacte normal deschise.

Dacă cu ajutorul a două contacte x și y realizăm un dipol serie (fig. 5.4), în care prin cercuri pline am marcat

Fig. 5.4

punctele de legătură dintre contacte și polii dipolului (nodurile grafului), iar prin x și y contactele respective (arcele grafului), atunci evident conductivitatea dipolului din fig. 5.4 este funcție de starea contactelor x și y.

Fie $z = f(x, y)$. Convenim ca variabila z să ia valoarea 1 dacă dipolul conduce și valoarea 0 în caz contrar. Valorile lui z în funcție de x și y sînt date în tabelul 5.1, de unde rezultă că dipolul serie realizează funcția **și**, $z = xy$.

Tabelul 5.1

x	y	xy	$x+y$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Dacă considerăm dipolul paralel obținut cu ajutorul contactelor x și y (fig. 5.5) și notăm cu z funcția $f(x, y)$ care exprimă conductivitatea acestui dipol, valorile

funcției f sînt date în tabelul 5.1, adică dipolul paralel realizează funcția **sau**, $z = x + y$.

Convenim să asociem un graf unei funcții booleene, în felul următor : considerăm o expresie a funcției date și

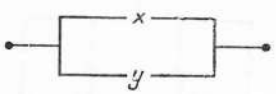

Fig. 5.5

convenim să numerotăm semnele · care apar în expresia respectivă, într-o ordine arbitrară, de exemplu în ordinea în care apar în expresie, numărîndu-le de la stînga la dreapta. Dacă unele semne · nu sînt scrise, dar sînt necesare, se subînțeleg și se numerotează. Extremitatea din stînga a expresiei și extremitatea din dreapta se numerotează de asemenea. Numerele asociate acestora se vor numi polii dipolului reprezentativ. Parantezele se numerotează și ele, astfel : parantezele care înlocuiesc semnul · poartă numărul asociat semnului respectiv ; parantezele care nu înlocuiesc semne · nu se numerotează ; mai multe paranteze succesive poartă același număr. Semnelor + li se asociază cîte o pereche de numere (a, b) după următoarea regulă : a este numărul primei paranteze de închidere din dreapta semnului + sau numărul polului din dreapta, dacă astfel de paranteze nu există iar b este numărul celei mai apropiate paranteze de deschidere din stînga semnului + sau numărul polului din stînga, dacă astfel de paranteze nu există. Numerele asociate în acest fel funcției booleene vor constitui nodurile grafului. Două noduri n_i, n_j vor fi unite printr-un arc marcat cu litera x, dacă în expresia considerată numerele n_i și n_j sînt vecine cu variabila x.

Definiție. Reprezentările obținute prin procedeul precedent vor fi numite *reprezentări prin scheme de contacte serie-paralel.*

În general, o funcție booleană admite mai multe reprezentări serie-paralel. Problema găsirii schemelor care realizează o funcție booleană cu număr minim de contacte va face obiectul unui capitol separat. În general, schemele

serie-paralel nu realizează acest minim, fiind necesară introducerea schemelor cu punți, care vor fi de asemenea tratate ulterior.

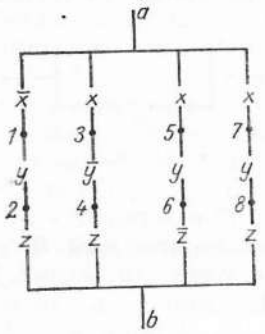

Fig. 5.6

Vom da în continuare un exemplu de reprezentare prin scheme serie-paralel a funcțiilor booleene.

Exemplu. Să considerăm funcția

$$f(x, y, z) = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz.$$

Vom asocia semnelor • și + numere după regula dată. Pentru poli asociem literele a și b, iar pentru semnele • numerele naturale, de la stînga spre dreapta. Neexistînd paranteze, semnele + vor primi toate perechile (b, a).

Vom rescrie funcția $f(x, y, z)$, înlocuind semnele cu numerele asociate și neglijînd parantezele nemarcate:

$$f(x, y, z) = a\bar{x}\,1\,y\,2\,z\,b\,a\,x\,3\,\bar{y}\,4\,z\,b\,a\,x\,5\,y\,6\,\bar{z}\,ba\,x\,7y\,8\,zb.$$

În fig. 5.6 este dat dipolul reprezentativ, construit după regula dată.

Dacă s-ar fi plecat de la expresia simplificată a funcției $f(x, y, z) = x(y + z) + yz$, s-ar fi obținut expresia $f(x, y, z) = a\,x\,1yb\,1\,zbay\,2\,zb$ și schema din fig. 5.7.

5.4. Reprezentarea prin scheme de operatori

O altă reprezentare a funcțiilor booleene este reprezentarea secvențială, prin scheme de operatori.

Convenim să reprezentăm în plan un operator printr-un semicerc, în interiorul căruia este trecut semnul sau denumirea operatorului, care are una sa umai multe intrări și o ieșire. În fig. 5.8, *a* este desenat un operator $\mathscr{V}_1$, unar, în care x este variabila de intrare, iar y rezultatul aplicării operatorului $\mathscr{V}$ variabilei x, $y = \mathscr{V}_1(x)$.

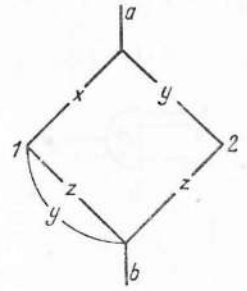

Fig. 5.7

În general, numim operator n-ar o funcție booleană de n variabile $y = f(x_1, x_2, \ldots, x_n)$. Mulțimea operatorilor utilizați în practică este determinată de posibilitatea realizării acestora prin elemente fizice.

Exemple de operatori:

1) Operatorul negație reprezintă funcția care aplică orice variabilă x în $\bar{x}$. Dacă-l notăm prin $\mathscr{V}_1$, atuci $\mathscr{V}_1(x) = \bar{x}$.

2) Operatorul •, care este un operator binar, îl vom nota în cele ce urmează cu $\mathscr{V}_2$. Vom avea: $\mathscr{V}_2(x, y) = xy$.

3) Operatorul +, care este de asemenea operator binar, îl vom nota cu $\mathscr{V}_3$. Vom avea: $\mathscr{V}_3(xy) = x + y$.

Ținînd seama de comutativitatea, asociativitatea și distributivitatea operațiilor •, +, vom putea scrie:

$$\left.\begin{aligned}\mathscr{V}_2(x, y) &= \mathscr{V}_2(y, x)\\ \mathscr{V}_3(x, y) &= \mathscr{V}_3(y, x)\end{aligned}\right\}\ \text{comutativitatea;}$$

$$\left.\begin{aligned}\mathscr{V}_2(\mathscr{V}_2(x, y), z) &= \mathscr{V}_2(x, \mathscr{V}_2(y, z))\\ \mathscr{V}_3(\mathscr{V}_3(x, y), z) &= \mathscr{V}_3(x, \mathscr{V}_3(y, z))\end{aligned}\right\}\ \text{asociativitatea;}$$

$$\left.\begin{aligned}\mathscr{V}_2(x, \mathscr{V}_3(y, z)) &= \mathscr{V}_3(\mathscr{V}_2(x, y), \mathscr{V}_2(x, z))\\ \mathscr{V}_3(x, \mathscr{V}_2(y, z)) &= \mathscr{V}_2(\mathscr{V}_3(x, y), \mathscr{V}_3(x, z))\end{aligned}\right\}\ \text{distributivitatea.}$$

Pe baza acestor proprietăți vom conveni să extindem operatorii b), c), $\mathscr{V}_2$ și $\mathscr{V}_3$ la mai multe variabile, folosind formulele de recurență (fig. 5.8).

$$\mathscr{V}_2(x_1, x_2, \ldots, x_n) = \mathscr{V}_2(\mathscr{V}_2(x_1, x_2, \ldots, x_{n-1}), x_n),$$

$$\mathscr{V}_3(x_1, x_2, \ldots, x_n) = \mathscr{V}_3(\mathscr{V}_3(x_1, x_2, \ldots, x_{n-1}), x_n).$$

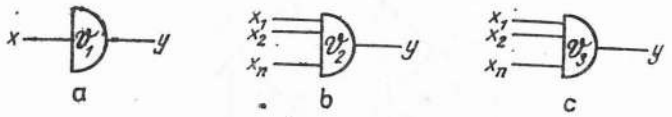

Fig. 5.8

După cum se vede, un operator este urmat de o pereche de paranteze, în interiorul cărora sînt înscrise variabilele, care pot fi la rîndul lor operatori. În continuare, vom numi aceste paranteze parantezele operatorului dat.

Convenim să asociem un graf secvențial, sau cu operatori, unei funcții booleene, în felul următor: vom scrie funcția booleană cu ajutorul operatorilor $\mathscr{V}_1, \mathscr{V}_2, \ldots, \mathscr{V}_n$ dați. În particular, putem scrie funcțiile booleene cu trei operatori: $\mathscr{V}_1$, $\mathscr{V}_2$, $\mathscr{V}_3$ sau chiar și cu un singur operator, operatorul **nici**, pe care-l vom nota cu $\mathscr{V}_4$; $\mathscr{V}_4(\bar{x}, \bar{y}) = xy$. Vom desena în plan mulțimea operatorilor care intervin și vom stabili intrările fiecărui operator, astfel: intrările în operatorul $\mathscr{V}$ sînt ieșirile din toți operatorii incluși în parantezele operatorului $\mathscr{V}$ și care nu sînt incluși în parantezele nici unui alt operator sau variabilele care îndeplinesc aceeași condiție. Se va obține o schemă arborescentă în care operatorii care apar primii începînd de la stînga vor fi desenați în partea dreaptă a figurii, iar operatorii care apar în aceeași paranteză vor apărea pe aceeași verticală.

Exemplu. Funcția $f(x, y, z) = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$ se exprimă cu ajutorul operatorilor $\mathscr{V}_1$, $\mathscr{V}_2$, $\mathscr{V}_3$ astfel:

$$f(x, y, z) = \mathscr{V}_3(\mathscr{V}_2(\mathscr{V}_1(x), y, z), \mathscr{V}_2(x, \mathscr{V}_1(y), z), \mathscr{V}_2(x, y, \mathscr{V}_1(z)), \mathscr{V}_2(x, y, z))$$

și aplicînd regula dată se obține schema din fig. 5.9.

Dacă am fi plecat de la expresia $f(x, y, z) = x(y+z) + yz$ a funcției, s-ar fi obținut expresia de operatori $f(x, y, z) = \mathscr{V}_3(\mathscr{V}_2(x, \mathscr{V}_3(y, z)), \mathscr{V}_2(y, z))$ are conduce la schema din fig. 5.10, care este numai cu patru operatori în loc de opt ca în fig. 5.9.

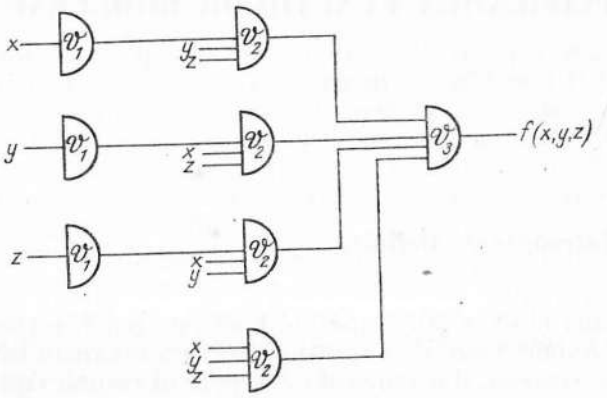

Fig. 5.9.

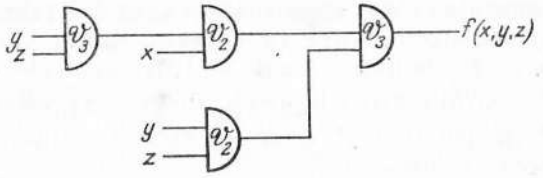

Fig. 5.10

SIMPLIFICAREA FUNCȚIILOR BOOLEENE

6.1. Introducere. Definiții

După cum se știe, funcțiile booleene pot fi scrise printr-un număr mare de expresii. Pentru a compara între ele aceste expresii, din punctul de vedere al complexității lor, este necesar să se stabilească anumite criterii. În acest scop a fost introdusă noțiunea de cost asociată unei expresii date. Pot fi alese funcții cost diferite, ținînd seama, de exemplu, de costul efectiv al elementelor tehnice utilizate pentru realizarea funcției booleene, sau de alte criterii. Funcțiile cost cele mai utilizate sînt următoarele :

C_1. Numărul total de litere cuprinse în expresie.

C_2. Numărul total de operatori sau de contacte, cu care se realizează expresia.

C_3. Numărul total de intrări în schemele cu operatori care realizează funcțiile date.

Trebuie să atragem atenția că aceste criterii nu sînt suficiente pentru obținerea unor reprezentări optimale ale funcțiilor prin scheme electrice. Spre exemplu, o expresie a unei funcții booleene care conține cel mai mic număr de litere poate conduce la scheme cu număr mai mare de contacte decît o expresie cu număr mai mare de litere. Numărul de elemente dintr-o reprezentare a funcției depinde nu numai de expresia de la care se pleacă, ci și de metoda de sinteză folosită, după cum vom demonstra în cap. 9. Criteriile C_2 și C_3 nu sînt potrivite pentru compararea expresiilor unei funcții date, la fel cum criteriul C_1 nu este potrivit pentru compararea între ele a schemelor care realizează funcția respectivă. În consecință, vom folosi criteriul C_1 pentru aprecierea complexității expresiilor și criteriile C_2 și C_3 pentru aprecierea complexității schemelor.

Definiția 1. Operația de trecere de la o expresie a unei funcții booleene la o expresie cu un număr mai redus de litere se va numi *simplificare*. Scopul operației de simplificare este de a obține expresiile cu număr minim de litere.

Un rol important în operația de simplificare îl joacă determinarea implicanților primi și a expresiilor care conțin numai implicanți primi.

Se ridică o problemă : cum este mai bine să se procedeze : să se caute expresiile cu număr minim de implicanți primi sau să se utilizeze expresia care conține toți implicanții primi ? Se pare că pînă acum s-a mers de regulă pe prima cale.

În această lucrare s-a ajuns la un rezultat aparent paradoxal și anume, folosind metoda de sinteză din cap. 9, expresiile care conțin toți implicanții primi conduc la reprezentări cu număr de elemente mai mic decît celelalte expresii.

Definiția 2. Expresia unei funcții booleene care este suma tuturor implicanților săi primi va fi denumită *expresie caracteristică*. Oricare altă expresie care conține numai implicanți primi va fi denumită *expresie redusă* iar expresiile care conțin un număr minim de implicanți primi se vor numi *expresii minimale*.

Metodele de simplificare ale funcțiilor booleene se pot împărți în două categorii : metode tabelare și metode algebrice. În prima categorie se încadrează metoda diagramei lui Veitch, metoda diagramei lui Karnaugh și și altele. În continuare nu ne vom ocupa de metodele tabelare, întrucît ne interesează în primul rînd metodele algebrice, programabile pe calculator. Dintre metodele de simplificare vom prezenta metoda lui Quine-McCluskey, metoda consensurilor a lui Tison și unele metode date de autor, metoda celui mai mare divizor comun și o metodă

similară cu metoda Quine-McCluskey care se pretează bine programării pe calculator. De asemenea, vor fi prezentate generalizările metodelor pentru funcțiile de variabile polivalente.

Simplificarea funcțiilor se face în două etape. Într-o primă etapă se caută expresia caracteristică iar în a doua se caută expresiile minimale, pornind de la expresia caracteristică. În acest capitol vom prezenta metode de rezolvare pentru amîndouă etapele de simplificare. Pentru ușurința exprimării metodelor referitoare la găsirea funcției caracteristice vor fi denumite *metode de simplificare*, iar cele referitoare la obținerea expresiilor minimale, metode de *minimizare*.

6.2. Metode de simplificare bazate pe compunerea vecinilor

6.2.1. *Metoda Quine-McCluskey*

Definiția 1. Doi implicanți ax și $a\bar{x}$ ai unei funcții booleene care diferă numai prin forma unei variabile se numesc *vecini*. Operația de simplificare dată de egalitatea $ax + a\bar{x} = a$ se va numi operația de *compunere a vecinilor*. Metoda Quine-McCluskey se bazează pe posibilitatea simplificării funcțiilor booleene prin compunerea succesivă a vecinilor.

În esență această metodă constă în efectuarea sistematică și completă a compunerilor de vecini.

Fie $\mathbf{M} = \{M_1, M_2, \ldots, M_m\}$ mulțimea constituenților unității ai funcției $f(x_1, x_2, \ldots, x_n)$. Fiecărui constituent al unității $M_i = x_1^{a_{i1}} x_2^{a_{i2}} \ldots x_n^{a_{in}}$ îi asociem numărul $q_i = \sum_{j=1}^{n} a_{ij}$ și vectorul $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$. Notăm cu $\mathbf{Q}$ mulțimea numerelor q_i și cu $\mathbf{A}$ mulțimea vectorilor $\mathbf{a}_i$:

$$\mathbf{Q} = \{q_1, q_2, \ldots, q_m\}, \quad \mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}.$$

Fie $\mathbf{P} = \{p_1, p_2, \ldots, p_r\}$ mulțimea numerelor distincte din $\mathbf{Q}$. Fiecărui număr $p_i \in \mathbf{P}$ îi asociem o submulțime $\mathbf{A}_j \in \mathbf{A}$ astfel:

$$(\mathbf{a}_j, \mathbf{a}_k \in \mathbf{A}_i) \Rightarrow (q_j = q_k = p_i).$$

Prin acest procedeu se obține o partiție a lui $\mathbf{A}$.

Fiecare clasă $\mathbf{A}_i$ a partiției se bucură de propietatea că vectorii care o compun au p_i componente diferite de 0. Se numesc *vecine* două clase $\mathbf{A}_i$ și $\mathbf{A}_k$ pentru care $|p_i - p_k| = 1$.

Constituienții unității vecini se găsesc în clase vecine, de unde rezultă că în cazul folosirii unei metode de căutare a vecinilor această căutare se reduce numai la clasele vecine.

Descrierea algoritmului. 1. Se asociază fiecărui constituent al unității M_i vectorul $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$.

2. Se asociază vectorilor $\mathbf{a}_i$ numerele $q_i = \sum_{j=1}^{n} a_{ij}$.

3. Se ordonează vectorii $\mathbf{a}_i$ în ordinea crescătoare a numerelor q_i și se grupează în clasele $\mathbf{A}_i$ vectorii care au același număr asociat p_i.

4. Se caută vecinii comparînd implicații din două grupe vecine și se aplică operația de compunere pentru toate perechile de implicanți vecini, marcînd de exemplu printr-o liniuță variabilele care dispar prin compunere.

5. Rezultatul compunerii se înscie într-un nou tabel după regula folosită la pașii 2 și 3.

6. Se repetă pașii 4 și 5 la noul tabel pînă se ajunge la un tabel ireductibil față de operația de compunere, cînd procesul încetează.

Exemplu. Să se simplifice funcția

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 x^0 + x_1^0 x_2^0 x_3^0 x_4 + x_1 x_2^0 x_3^0 x_4 + x_1^0 x_2 x_3^0 x_4 + x_1^0 x_2 x_3 x_4 + x_1 x_2 x_3 x_4.$$

Aplicînd pașii 1, 2 și 3, se obține rezultatul din tabelul 6.1, *a*.

Tabelul 6.1.

x_1	x_2	x_3	x_4
1	0	0	0
0	0	0	1
1	0	0	1
1	1	0	1
0	1	1	1
0	1	1	1

a

x_1	x	x_3	x_4
1	0	0	—
—	0	0	1
0	—	0	1
0	1	—	1
—	1	1	1
.			

b

Aplicînd pașii 4 și 5, se obține rezultatul din tabelul 6.1, *b*, care este ireductibil față de operația de compunere. Rezultatul simplificării este deci expresia caracteristică a funcției

$$f(x_1, x_2, x_3, x_4) = x_1x_2^0x_3^0 + x_2^0x_3^0x_4 + x_1^0x_3^0x_4 + x_1x_2x_4 + x_2x_3x_4.$$

6.2.2. *Algoritm de minimizare*

Fie $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ mulțimea implicanților primi ai funcției booleene f și $\mathbf{M} = \{M_1, M_2, \ldots, M_m\}$ mulțimea constituenților unității incluși în f. Fiecărui constituent al unității M_i i se va asocia un vector boolean $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ după următoarea regulă: $(a_{ik} = 1) \Leftrightarrow (M_i \Rightarrow P_k)$, $a_{ik} = 0$ în caz contrar. Fiecărui constituent M_i al unității îi asociem apoi suma $S_i = \sum_{k=1}^{n} a_{ik}P_k$.

Considerăm funcția $F(P_1, P_2, \ldots, P_n) = \prod_{i=1}^{n} S_i$. Fie $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_r\}$ mulțimea implicanților primi ai funcției F și $\mathbf{P}^s = \{P_1^s, P_2^s, \ldots, P_{qs}^s\}$ implicanții primi ai funcției f care apar în Q_s, $s \in \mathbf{I}_r = \{1, 2, \ldots, r\}$.

Lemă. Pentru orice $s \in \mathbf{I}_r$, $f = \sum_{j=1}^{n} P_j^s$ este o expresie minimală a funcției f.

Demonstrație. Din $P_j^s \in \mathbf{P}$ rezultă $P_j^s \Rightarrow f$, $(\forall)\, s \in \mathbf{I}_r$ și deci $\sum_{j=1}^{q_s} P_j^s \Rightarrow f$. Fie apoi $M_i \in \mathbf{M}$ un constituent oarecare al unității inclus în f și $S_i = \sum_{k=1}^{n} a_{ik}P_k$ suma asociată lui M_i. Deoarece Q_s este implicant prim al funcției F, există $P_j^s \in \mathbf{P}^s$ care are coeficient egal cu 1 în suma S_i. Fie t indicele acestui coeficient în mulțimea $\mathbf{P}$, deci $P_j^s = P_t$. Dar atunci, conform presupunerii, $M_i \Rightarrow P_t = P_j^s$ și $M_i \Rightarrow \sum_{i=1}^{q_s} P_j^s$. Întrucît M_i a fost ales arbitrar în $\mathbf{M}$, rezultă că implicația

precedentă este adevărată pentru orice $M_i \in \mathbf{M}$ și deci $\mathbf{f} \Rightarrow \sum_{i=1}^{q_s} P_j^s$, lema fiind astfel complet demonstrată.

Lema precedentă ne conduce la algoritmul de minimizare.

Descrierea algoritmului. Se presupun date mulțimea $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ a implicanților primi și mulțimea $\mathbf{M} = \{M_1, M_2, \ldots, M_m\}$ a constituenților unității incluși în funcția dată.

1. Se asociază constituenților unității M_i, $i \in \mathbf{I}_m$, vectorii booleeni $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, după regula $(a_{ik} = 1) \Leftrightarrow (M_i \Rightarrow P_k)$, $a_{ik} = 0$ în caz contrar.

2. Se asociază constituenților M_i sumele $S_i = \sum_{k=1}^{n} a_{ik}P_k$.

3. Se construiește funcția $F(P_1, P_2, \ldots, P_n) = \prod_{k=1}^{n} S_i$.

4. Se simplifică funcția F și se obține mulțimea $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_r\}$ a implicanților săi primi.

5. Se asociază implicanților primi Q_s ai funcției F submulțimile $\mathbf{P}^s = \{P_1^s, P_2^s, \ldots, P_{qs}^s\}$ de implicanți primi ai funcției f, unde P_j^s, $j \in \{1, 2, \ldots, q_s\}$, sînt implicanții primi ai funcției f care apar în Q_s.

Expresiile $f = \sum_{i=1}^{q_s} P_j^s$ sînt expresii reduse ale funcției f.

Consecință. *Numărul total de expresii reduse ale funcției f este egal cu numărul de implicanți primi ai funcției F asociate.*

Se pot pune două probleme de minimizare. Prima constă în căutarea expresiilor care conțin un număr minim de implicanți primi, pentru rezolvarea căreia este suficient să luăm expresiile corespunzătoare elementelor lui Q de lungime minimă. A doua se reduce la căutarea din mulțimea expresiilor reduse pe cele cu număr minim de variabile. Cele două probleme sînt diferite, adică nu întotdeauna expresiile cu număr minim de variabile au și numărul minim de implicanți primi.

6.3. Metoda hiperfețelor maximale

6.3.1. *Vectori booleeni și numere convenționale asociate implicanților*

Metoda Quine-McCluskey prezentată în 6.2 se pretează foarte bine pentru calcule manuale și pentru funcții cu număr redus de variabile. Pentru simplificarea cu ajutorul calculatorului electronic prezintă dezavantajul că începînd cu pasul doi unii implicanți se obțin pe mai multe căi; numărul de posibilități de obținere a implicanților în pasul k al procesului de simplificare crește repede cu numărul pasului respectiv. Pe de altă parte, trebuie rezolvată problema identificării locurilor marcate, ceea ce conduce la complicarea algoritmului de calcul.

Se pune deci problema evitării înregistrării multiple a implicanților în cursul procesului de simplificare și reducerii complexității algoritmului de calcul. În continuare vom prezenta o metodă care înlătură aceste inconveniente [31].

Convenim să scriem implicanții unei funcții booleene de n variabile sub forma de vectori booleeni cu $2n$ componente, prin scrierea consecutivă a doi vectori booleeni de cîte n componente, după cum se va arăta mai departe. Fie $\mathscr{P}_a(\mathbf{Y}) = y_1^{a_1} y_2^{a_2} \dots y_p^{a_p}$ un implicant al funcției boolene de n variabile $f(x_1, x_2, \dots, x_n)$, $\mathbf{Y} = \{y_1, y_2, \dots, y_p\}$ mulțimea variabilelor care apar în $\mathscr{P}_a(\mathbf{Y})$ și $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ mulțimea vaiabilelor lui f. Lui $\mathbf{Y}$ i se asociază perechea de vectori $(\mathbf{u}, \mathbf{v})$ după următoarea regulă:

a) $\mathbf{u} = (u_1, u_2, \dots, u_n)$, unde $u_i = 0$ pentru $x_i \in \mathbf{Y}$ și $u_i = 1$ în caz contrar;

b) $\mathbf{v} = (v_1, v_2, \dots, v_n)$, unde $v_i = a_i$ pentru $x_i \in \mathbf{Y}$ și $v_i = 0$ în caz contrar.

Rezultă că vectorul $\mathbf{u}$ marchează variabilele care lipsesc din $\mathbf{Y}$.

Definiția 1. Vectorul boolean $\mathbf{w} = (\mathbf{u}, \mathbf{v})$ obținut prin scrierea succesivă a vectorilor $\mathbf{u}$ și $\mathbf{v}$ se va numi *vectorul boolean asociat implicantului* $\mathbf{Y}$ sau codul bivalent al lui $\mathbf{Y}$, noțiune introdusă în 4.4.2.

Uneori pentru simplificarea scrierii se folosesc în locul vectorilor booleeni numerele convenționale asociate, definite astfel:

Definiția 2. Dacă $\mathbf{a} = (a_1, a_2, \dots, a_n)$ este un vector boolean, atunci numărul $N(\mathbf{a}) = \sum_{i=1}^{n} a_{n-i+1}\, 2^{i-1}$ va fi denumit *numărul convențional al vectorului* $\mathbf{a}$. În consecință, unui implicant oarecare i se poate asocia perechea de numere convenționale $N(\mathbf{u})$, $N(\mathbf{v})$ corespunzătoare vectorilor booleeni asociați $\mathbf{u}$ și $\mathbf{v}$.

În general, unui implicant al unei funcții de n variabile i se poate asocia fie o pereche de numere convenționale (a, b), fie un singur număr convențional, $N(\mathbf{w})$, numărul convențional al lui $\mathbf{w}$.

6.3.2. *Operații cu numere convenționale*

Fie $\mathbf{A}$, $\mathbf{B}$, și $\mathbf{C}$ trei vectori binari iar a, b și c numerele convenționale asociate. În mulțimea numerelor convenționale se introduce o operație notată cu $\oplus$ definită astfel: $a \oplus b = c \Leftrightarrow \mathbf{A} + \mathbf{B} = \mathbf{C}$, $\oplus$ reprezentînd suma modulo 2 a vectorilor binari $\mathbf{A}$ și $\mathbf{B}$.

Fie (a, b) și (x, y) doi implicanți ai unei funcții booleene exprimați prin numerele convenționale. Între implicanții (a, b) și (x, y) vom introduce operația $\oplus$ definită astfel: $(a, b) \oplus (x, y) = (a \oplus x, b \oplus y)$. Doi implicanți (a, b) și (x, y) vecini se bucură de proprietatea: $a \oplus x = 0$, respectiv $a = x$, și $b \oplus y = 2^u$, respectiv b și y au o singură componentă diferită.

Fie (a, b) și (a, c) doi implicanți vecini, astfel încît $b < c$ și $b \oplus c = 2^u$. Între doi implicanți de această formă introducem o nouă operație notată cu semnul $\boxplus$ definită astfel:

$$(a, b) \boxplus (a, c) = (a + (b \oplus c), b), \quad b < c.$$

Se poate demonstra

Propoziție. Dacă (a, b) și (a, c) sînt implicanți vecini și dacă $b < c$, atunci $(a, b) \boxplus (a, c)$ coresunde implicantului obținut prin compunerea lui (a, b) cu (a, c). Operația $\boxplus$ va fi numită, din această cauză, *operația de compunere a vecinilor*.

Teorema următoare este esențială pentru descrierea algoritmului [31]:

Teoremă. *Dacă implicantul (a, b) se obține în pasul k al procesului de simplificare dintr-o mulțime dată de perechi vecine $(a_1, b)\boxplus(a_1, b_1) = (a, b)$, $(a_2, b)\boxplus(a_2, b_2) = (a, b)$, ..., $(a_r, b)\boxplus(a_r, b_r) = (a, b)$ astfel încît $b \oplus b_j = 2^{i_j}$, $b < b_j$, $(\forall)$ $j = 1, 2, \ldots, r$, atunci există un indice j_0 astfel încît*

$$(a_j, b) > (2^{i_j}, 0), j \neq j_0, \ (a_j, b) < (2^{i_j}, 0), j = j_0.$$

Sîntem acum în măsură să descriem algoritmul de îmbunătățire a metodei lui Quine-McCluskey [31].

6.3.3. *Descrierea algoritmului*

1. Dacă $f(x_1, x_2, \ldots, x_n)$ este funcția booleană dată, aceasta se scrie sub forma canonică disjunctivă și se asociază fiecărui constituent al unității o pereche de numere convenționale după cum s-a arătat mai înainte. Aceste perechi vor fi de forma $(0, i)$, unde i sînt numerele convenționale asociate constituenților unității, iar primul număr zecimal din fiecare pereche este zero deoarece în constituenții unității apar toate variabilele. Perechile $(0, i)$ asociate funcției se înscriu într-un tabel $\mathbf{M}_0$.

2. Se construiește un șir de mulțimi de perechi $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_2$, ..., astfel încît trecerea de la mulțimea $\mathbf{M}_k$ la mulțimea $\mathbf{M}_{k+1}$ se face astfel:

— se aplică perechilor de implicanți (a, b), (c, d) din $\mathbf{M}_k$ operația $\oplus$;

— dacă $(a, b) \oplus (c, d) = (0, 2^u)$, se aplică implicanților (a, b) și (a, d) operația $\boxplus$ și rezultatul se înscrie în $\mathbf{M}_{k+1}$;

— un implicant (a, b) din $\mathbf{M}_k$ pentru care nu există nici un alt implicant din $\mathbf{M}_k$ astfel încît $(a, b) \oplus (a, c) = (0, 2^u)$ este un implicant prim și se înscrie într-un tabel

de implicanți primi, **T**. Tabelul **T** se inițializează în valoarea $\mathbf{T} = \emptyset$.

3. Procesul încetează cînd $\mathbf{M}_k$ este vid.

Exemplu. Să se simplifice funcția booleană, dată de mulțimea de numere convenționale $\mathbf{F} = \{0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14\}$.

1) $\mathbf{M}_0 = \{(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 6), (0, 7), (0, 8), (0, 9), (0, 10), (0, 11), (0, 14)\}$.

1.2) $\mathbf{M}_1 = \{(1, 0), (2, 0), (4, 0), (8, 0), (2, 1), (8, 1), (1, 2), (4, 2), (8, 2), (4, 3), (8, 3), (2, 4), (1, 6), (8, 6), (1, 8), (2, 8), (2, 9), (1, 10), (4, 10)\}$.
$\mathbf{T} = \emptyset$.

2.2) $\mathbf{M}_2 = \{(3, 0), (9, 0), (6, 0), (10, 0), (10, 1), (5, 2), (9, 2), (12, 2), (3, 8)\}$,
$\mathbf{T} = \emptyset$.

2.3) $\mathbf{M}_3 = \{(11, 0)\}$,
$\mathbf{T} = \{(12, 2), (6, 0), (5, 2)\}$.

2.4) $\mathbf{M}_4 = \emptyset$,
$\mathbf{T} = \{(12, 2), (6, 0), (5, 2), (11, 0)\}$.

6.4. Metoda consensurilor

6.4.1. *Definiții. Proprietăți*

Un alt neajuns al metodei Quine-McCluskey, în afara celor puse în evidență în 6.3, constă în faptul că sumplificarea trebuie făcută plecînd întotdeauna de la forma canonică disjunctivă a funcției date, nu de la o formă normală disjunctivă oarecare. Această condiție trebuie îndeplinită pentru evitarea scăpării unor implicanți primi. Îndeplinirea ei mărește volumul de calcule și dimensiunile problemelor de simplificare. Înlăturarea acestui neajuns se poate face bazînd simplificarea nu pe compunerea vecinilor ca în metoda Quine-McCluskey, ci pe implicația $ab \Rightarrow ax + b\bar{x}$, pe care o vom denumi *implicația consensului*, produsul ab fiind denumit *consensul implicanților* ax și $b\bar{x}$. Să remarcăm că implicația consensului este o generalizare a relației $a \Rightarrow ax + a\bar{x}$; într-adevăr, se poate verifica ușor că pentru $a = b$ implicația consensului se reduce la o relație de echivalență $a = ax + a\bar{x}$.

O metodă bazată pe implicația consensului a fost dată de Tison.

Evident, pentru ca implicația $ab \Rightarrow ax + b\bar{x}$ să nu fie banală este necesar ca produsul ab să fie nenul, adică a și b să conțină variabile monoforme, înțelegînd prin variabile monoforme variabilele care apar sub o singură formă, fie directă, fie negată, atît în a cît și în b, dacă apar în amîndouă. Ținînd seama de cele arătate rezultă că metoda lui Tison este de fapt o generalizare a metodei Quine-McCluskey.

Noțiunea de consens poate fi la rîndul ei generalizată, trecînd de la consensul a doi termeni la consensul a n termeni.

Definiția 1. Un termen a_0 este *consensul termenilor* $a_1, a_2, \ldots, a_n$ dacă $a_0 \Rightarrow \sum_{i=1}^{n} a_i$ și dacă $a_0 \not\Rightarrow \sum_{i=1}^{n}{}' a_i$, unde $\sum_{i=1}^{n}{}' a_i$ se obține din $\sum_{i=1}^{n} a_i$ prin eliminarea, cel puțin, a unui termen oarecare.

Consecință. *Consensul termenilor* $a_1, a_2, \ldots, a_n$, *dacă există, este diferit de zero.*

Într-adevăr, $a_0 = 0 \Rightarrow (a_0 \Rightarrow \sum_{i=1}^{n}{}' a_i)$ contrar definiției 1.

Convenim să facem notația $a_0 = \mathscr{C}\{a_1, a_2, \ldots, a_n\}$.

Proprietatea 1. *Dacă* $a_0 = \mathscr{C}\{a_1, a_2, \ldots, a_n\}$, *atunci există relațiile* $a_0 \sum_{i=1}^{n} a_i = a_0$, $a_0 \sum_{i=1}^{n}{}' a_i \neq a_0$.

Demonstrație. $a_0 \Rightarrow \sum_{i=1}^{n}{}' a_i \Leftrightarrow \bar{a}_0 + \sum_{i=1}^{n}{}' a_i = 1 \Rightarrow a_0 \sum_{i=1}^{n} a = a_0$;

$$a_0 \not\Rightarrow \sum_{i=1}^{n}{}' a_i \Leftrightarrow \bar{a}_0 + \sum_{i=1}^{n}{}' a_i \neq 1 \Rightarrow a_0 \sum_{i=1}^{n} a_i \neq a_0.$$

Proprietatea 2. *Pentru orice* $x \in a_0$, $a_0 = \mathscr{C}\{a_1, a_2, \ldots, a_n\}$, $\bar{x} \notin \sum_{i=1}^{n} a_i$, *unde* $x \in a_0$ *are semnificația „*x *apare în monomul* a_0*“.*

Demonstrație. Notăm $A = \sum_{i=1}^{n} a_i$, $A' = \sum_{i=1}^{n}{}' a_i$ și presupunem că $\bar{x} \in A$. Există atunci a_i astfel încît $\bar{x} \in a_i$. Dar $x \in a_0$ și $\bar{x} \in a_i \Rightarrow a_0 a_i = 0$ și deoarece $A = A' + a_i$, se obține succesiv $a_0 A = a_0 = a_0 A' + a_0 a_i = a_0 A'$, contrar proprietății 1, ceea ce demonstrează proprietatea 2.

Proprietatea 3. *Fie* $A = \sum_{i=1}^{n} a_i$ și $a_0 = \mathscr{C}\{a_1, a_2, \ldots, a_n\}$. *Dacă* $x \in A$ *și* $\bar{x} \notin A$, atunci $a_0 x = a_0$.

Conform presupunerii se poate scrie $A = Dx + B$, unde D și B nu conțin pe x, cu $D \neq 0$ și $B \neq 1$ pentru ca x să apară efectiv în A. Ținînd seama de prima relație din proprietatea 1, se obține succesiv

$$a_0(Dx + B) = a_0 \Rightarrow \overline{a_0(Dx + B)}a_0 + a_0(Dx + B)\bar{a}_0 =$$

$$= 0 \Rightarrow \bar{a}_0 + (\bar{D} + \bar{x})\bar{B}a_0 = 0 \Rightarrow a_0\bar{D}\bar{B} + a_0\bar{x}\bar{B} = 0 \Rightarrow a_0\bar{x}\bar{B} = 0.$$

Se pot întîmpla următoarele cazuri:

1) $\bar{x}\bar{B} = 0 \Rightarrow \bar{x}B = \bar{x}$, ceea ce este imposibil întrucît $B \neq 1$ și nu conține pe $\bar{x}$.

2) $a_0\bar{B} = 0 \Rightarrow a_0 B = a_0$ contrar relației $a_0 A \neq a_0$, adevărată pentru $A = B$, după cum rezultă din proprietatea 1.

3) Rămîne $a_0\bar{x} = 0 \Rightarrow xa_0 = a_0 \Rightarrow x \in a_0$, ceea ce demonstrează proprietatea.

Proprietatea 4. *Fie* $a_i = b_i c_i$, *unde* b_i *conține variabilele monoforme din* $\sum_{i=1}^{n} a_i$ *cuprinse în* a_i, *iar* c_i *variabilele biforme din* $\sum_{i=1}^{n} a_i$ *cuprinse în* a_i.

Dacă există $a_0 = \mathscr{C}\{a_1, a_2, \ldots, a_n\}$, *atunci există relațiile* $\sum_{i=1}^{n} c_i = 1$, $\sum_{i=1}^{n}{}' c_i \neq 1$.

Demonstrație. Din $a_i = b_i c_i$, unde b_i conține variabilele monoforme din $\sum_{i=1}^{n} a_i$, se deduce ținînd seama de proprietatea 3 : $a_0 b_i = a_0 \Rightarrow a_0 a_i = a_0 c_i$. Apoi $a_0 \sum_{i=1}^{n} a_i = a_0 \Rightarrow$ $\Rightarrow \sum_{i=1}^{n} a_0 a_i = a_0 \Rightarrow \sum_{i=1}^{n} a_0 c_i = a_0 \Rightarrow a_0 \sum_{i=1}^{n} c_i = a_0 \Rightarrow \sum_{i=1}^{n} c_i = 1$, întrucît c_i conține variabilele diferite de a_0 pentru orice i.

La fel se arată că $\sum_{i=1}^{n}{}' c_i \neq 1$.

Definiția 3. Expresia $\sum_{i=1}^{n} c_i$ se numește *reziduul reuniunii* $\sum_{i=1}^{n} a_i$ în raport cu a_0. Termenul c_i este reziduul lui a_i în raport cu a_0.

Fie $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_r\}$, $\mathbf{Q} \subset \{a_1, a_2, \ldots a_n\}$, mulțimea termenilor a_i care conțin variabila x și $\mathbf{P} = \{P_1, P_2, \ldots, P_s\}$, $\mathbf{P} \subset \{a_1, a_2, \ldots, a_n\}$, mulțimea termenilor a_i care conțin variabila $\bar{x}$. Notăm prin Q_i' restul din Q_i după ce s-a înlăturat x și prin P_i' restul din P_i după ce s-a înlăturat $\bar{x}$.

Definiția 4. Q_i', $i \in \mathbf{I}_r$, se vor numi *reziduurile în raport* cu x iar P_i', $i \in \mathbf{I}_s$, reziduurile în raport cu $\bar{x}$.

Proprietatea 5. *Dacă* $Q_i' P_j' \neq 0$, *atunci* $Q_i' P_j' = \mathscr{C}\{Q_i, P_j\}$, $(\forall)\, i \in \mathbf{I}_r$, $(\forall)\, j \in \mathbf{I}_s$.

Demonstrație. $Q_i'\ P_j' \Rightarrow Q_i'\, x + P_j' \bar{x} = Q_i + P_j$. Apoi $Q_i'\ P_j' \not\Rightarrow Q_i$. Într-adevăr, $Q_i'\ P_j' \Rightarrow Q_i \Leftrightarrow Q_i'\ P_j'\ \bar{Q}_i = 0$. Dar $Q_i'\ \bar{Q}_i = Q_i'\, \bar{x} \neq 0$ și la fel $P_j'\ \bar{Q}_i = P_j'\ \bar{Q}_i + P_j'\, \bar{x} \neq 0$, adică implicația $Q_i'\ P_j' \Rightarrow Q_i$ nu are loc. La fel se arată că nici implicația $Q_i'\ P_j' \Rightarrow P_j$ nu are loc, adică $Q_i'\ P_j'$ este efectiv consensul $\mathscr{C}\{Q_i, P_j\}$.

Definiția 5. Consensurile astfel obținute vor fi denumite *consensuri în raport cu* x. Uneori pentru a scoate în evidență acest lucru vom scrie $\mathscr{C}_x\{Q_i, P_j\}$.

Proprietatea 6. *Doi termeni* Q_i *și* P_j *admit consens numai în raport cu o singură variabilă.*

Demonstrație. Dacă există $\mathscr{C}_x\{Q_i, P_j\}$, atunci de exemplu $Q_i = Q_i'\, x$ și $P_j = P_j'\, \bar{x}$ cu $Q_i'\ P_j' \neq 0$, adică Q_i și P_j nu conțin altă variabilă biformă în afară de x.

Această proprietate ne arată că, dacă există, consensul a doi termeni este unic, ceea ce ne permite să ignorăm variabila în raport cu care se face consensul.

Proprietatea 7. $\mathscr{C}\{Q_i, P_j\} = \mathscr{C}\{P_j, Q_i\}$.
Demonstrația rezultă imediat din definiția consensului.

Proprietatea 8. *Fie* $u = \sum_{i=1}^{n} P_i$, $v = \mathscr{C}\{P_1, P_2, \ldots, P_n\}$. *Dacă* $u \Rightarrow f$ *și* $u \neq 0$, *atunci* $v \Rightarrow f$.

Demonstrație. $(u \Rightarrow f) \Leftrightarrow (u\bar{f} = 0)$, $(v \Rightarrow u) \Leftrightarrow (v\bar{u} = 0)$. Se obține ecuația $u\bar{f} + v\bar{u} = 0$, care admite soluție pentru u, dacă $v\bar{f} = 0 \Leftrightarrow (v \Rightarrow f)$.

Caz particular. Pentru $f = u$ se obține $v \Rightarrow u$, adică implicația consensului.

Proprietatea 9. *Fie* $u = \mathscr{C}\{Q, R\}$, $v = \mathscr{C}(P, u)$ *și* $w = \mathscr{C}\{P, Q, R\}$; *atunci* $v \Rightarrow w$.

Demonstrație. Ținînd seama de proprietatea consensului, se obține succesiv : $u\bar{Q}\bar{R} = 0$, $v\bar{P}\bar{u} = 0 \Rightarrow u\bar{Q}\bar{R} + v\bar{P}\bar{u} = 0$, ecuație care are soluție pentru u cu condiția $v\bar{P}\bar{Q}\bar{R} = 0$. Dar și $w\bar{P}\bar{Q}\bar{R} = 0$, adică $(v + w)\bar{P}\bar{Q}\bar{R} = 0$. Deoarece w este maximal cu această proprietate, rezultă $v + w = w \Rightarrow$ $\Rightarrow (v\bar{w} = 0) \Leftrightarrow (v \Rightarrow w)$.

6.4.2. *Descrierea metodei*

1. Se determină variabilele biforme și se aranjează într-o ordine arbitrară $x_1, x_2, \ldots, x_p$.

2. Se stabilește tabelul reziduurilor în raport cu x_1 și $\bar{x}_1$ (se iau în considerare termenii care conțin efectiv pe x_1 și $\bar{x}_1$).

3. Se face produsul termen cu termen al reziduurilor în raport cu x_1 cu reziduurile în raport cu $\bar{x}_1$. Termenii diferiți de zero astfel obținuți constituie consensuri în raport cu x_1. Se adaugă aceste consensuri la termenii inițiali și după ce se elimină termenii care se absorb, rămîne o nouă mulțime de implicanți la care se aplică pașii 2 și 3 în raport cu o altă variabilă din mulțimea $x_1, x_2, \ldots, x_p$.

4. Procesul încetează după aplicarea pașilor 2 și 3 la toate variabilele biforme $x_1, x_2, x_3, \ldots, x_p$.

Exemplu. Să se determine expresia care conține toți implicanții primi ai funcției $f = x_1x_5 + x_2x_6 + x_3\bar{x}_7 + x_4\bar{x}_5\bar{x}_6x_7$.

1. Variabilele biforme sînt x_5, x_6 și x_7.

2.1. Reziduul în raport cu x_5 este x_1. Reziduul în raport cu $\bar{x}_5$ este $x_4\bar{x}_6x_7$.

3.1. Consensul în raport cu x_5 este $x_1x_4\bar{x}_6x_7$. Noua expresie a lui f este

$$f = x_1x_5 + x_2x_6 + x_3\bar{x}_7 + x_4\bar{x}_5\bar{x}_6x_7 + x_1x_4\bar{x}_6x_7.$$

2.2. Reziduul în raportul cu x_6 este x_2. Reziduurile în raport cu $\bar{x}_6$ sînt $x_4\bar{x}_5x_7$, $x_1x_4x_7$.

3.2. Consensurile în raport cu x_6 sînt $x_2x_4\bar{x}_5x_7$, $x_1x_2x_4x_7$. Noua expresie a lui f este

$$f = x_1x_5 + x_2x_6 + x_3\bar{x}_7 + x_4\bar{x}_5\bar{x}_6x_7 + x_1x_4\bar{x}_6x_7 + x_1x_2x_4x_7 + x_2x_4\bar{x}_5x_7.$$

2.3. Reziduurile în raport cu x_7 sînt $x_4\bar{x}_5x_6$, $x_1x_4\bar{x}_6$, $x_1x_2x_4$, $x_2x_4\bar{x}_5$. Reziduul în raport cu $\bar{x}_7$ este x_3.

3.3. Consensurile în raport cu x_7 sînt: $x_3x_4\bar{x}_5\bar{x}_6$, $x_1x_3x_4\bar{x}_6$, $x_1x_2x_3x_4$, $x_2x_3x_4\bar{x}_5$.

Noua expresie a lui f este

$$f = x_1x_5 + x_2x_6 + x_3\bar{x}_7 + x_4\bar{x}_5\bar{x}_6x_7 + x_1x_4\bar{x}_6x_7 + x_1x_2x_4x_7 + x_2x_4\bar{x}_5x_7 + x_3x_4\bar{x}_5\bar{x}_6 + x_1x_3x_4\bar{x}_6 + x_1x_2x_3x_4 + x_2x_3x_4\bar{x}_5,$$

care este expresia caracteristică a funcției respective.

6.4.3. *Relații de minimalitate*

Pentru obținerea expresiilor minimale nu se mai poate aplica algoritmul din 6.2, deoarece funcția nu este scrisă sub forma disjunctivă normală, ca reuniune a tuturor constituenților unității pe care-i cuprinde. Pentru eliminarea

acestui inconvenient vom da o metodă bazată pe teoria consensurilor. Înainte de aceasta vom prezenta relațiile și implicațiile de minimalitate.

Definiția 1. Fie $\mathbf{P} = \{P_1, P_2, \ldots, P_r\}$ mulțimea implicanților primi ai funcției f și $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_s\}$ o submulțime a lui $\mathbf{P}$ cu proprietatea $f = \sum_{i=1}^{s} Q_i$, $f \not\equiv \sum_{i=1}^{s}{}' Q_i$, unde $\sum_{i=1}^{s}{}' Q_i$ se obține din $\sum_{i=1}^{s} Q_i$ prin eliminarea cel puțin a unui Q_i. Vom numi aceste relații *relațiile de minimalitate.*

Definiția 2. O mulțime $\mathbf{Q}$, ale cărei elemente îndeplinesc relațiile de minimalitate, se spune că are *proprietatea de minimalitate.*

Prin minimizarea funcției f înțelegem determinarea expresiilor care îndeplinesc relațiile de minimalitate.

Lema 1. *Relațiile de minimalitate sînt echivalente cu următoarele implicații:*

$$P_j \Rightarrow \sum_{i=1}^{s} Q_i,\ (\forall)\ j \in \mathbf{I}_r,\ (\exists)\ j \in \mathbf{I}_r,\ P_j \not\equiv \sum_{i=1}^{s}{}' Q_i.$$

Demonstrație. 1) $Q_i \in \mathbf{P}$, $(\forall)\ i \in \mathbf{I}_s \Rightarrow \sum_{i=1}^{s} Q_i \Rightarrow f$.

2) $P_j \Rightarrow \sum_{i=1}^{s} Q_i$, $(\forall)\ j \in \mathbf{I}_r \Rightarrow \left(f \Rightarrow \sum_{i=1}^{s} Q_i \right)$:

Din 1) și 2) se deduce $f = \sum_{i=1}^{s} Q_i$.

3) $(\exists)\ j \in \mathbf{I}_r$, $P_j \not\equiv \sum_{i=1}^{s}{}' Q_i \Rightarrow f \neq \sum_{i=1}^{s}{}' Q_i$.

Definiția 3. Implicațiile din lema 1 vor fi denumite *implicații* de *minimalitate.* Notăm prin $\mathbf{R} = \mathscr{P}(\mathbf{P})$ mulțimea părților lui $\mathbf{P}$ și prin $\mathscr{C}(\mathbf{A})$ consensul termenilor din $\mathbf{A}$,

$\mathbf{A} \subset \mathbf{P}$. Fie $\mathbf{Q} \subset \mathbf{P}$ o submulțime a lui $\mathbf{P}$ cu proprietatea de minimalitate și $N_j = \{j \mid P_j \in \mathbf{P}\}$ mulțimea indicilor termenilor din $\mathbf{P}$. Fiecărui $\mathbf{P}$, $P_j \in \mathbf{P}$, $j \in N_i$, îi asociem o variabilă booleană x_j astfel încît $(x_j = 1) \Leftrightarrow P_j \in \mathbf{Q}$ și $x_j = 0$ în caz contrar. Pe de altă parte, fiecare termen $P_j \in \mathbf{P}$ realizează o partiție a lui $\mathbf{R}$ în două clase, $\mathbf{R}_j$ și $\bar{\mathbf{R}}_j$, $\mathbf{R}_j \cup \mathbf{R}_j = \mathbf{R}$, $\mathbf{R}_j \cap \mathbf{R}_j = \emptyset$, după regula : $\mathbf{R}_j^k \in \mathbf{R}_j$ dacă $\mathscr{C}(\mathbf{R}_j^k) = P_j$ și $\mathbf{R}_j^k \in \mathbf{R}_j$ dacă $\mathscr{C}(\mathbf{R}_j^k) \neq P_j$. Fie $\mathbf{K}_j = \{k \mid \mathbf{R}_j^k \in \mathbf{R}_j\}$ mulțimea indicilor k ai termenilor $\mathbf{R}_j^k \in \mathbf{R}_j$, $\mathbf{A}_j^k = \{x_i \mid P_i \in \mathbf{R}_j^k\}$ mulțimea variabilelor booleene x_j asociate termenilor P_i din $\mathbf{R}_j^k$ și $N_j^k = \{i \mid x_i \in \mathbf{A}_j^k\}$ mulțimea indicilor variabilelor x_i din $\mathbf{A}_j^k$.

Notăm $S_j^k = \prod_{i \in N_j^k} x_i$ produsul variabilelor x_i din $\mathbf{A}_j^k$ și $E_j = x_j + \sum_{k \in \mathbf{K}_j} S_j^k$ pentru orice $j \in N_j$. Funcției booleene f îi asociem expresia $E = \prod_{j \in N_j} E_j$.

Fie $X = x_1 x_2 \ldots x_s$ un implicant prim al funcției E și $\mathbf{Q}_i \in \mathbf{P}$, $i = 1, 2, \ldots, s$, termenii din $\mathbf{P}$ asociați variabilelor booleene $x_1, x_2, \ldots, x_s$.

Lema 2. *$X = 1 \Rightarrow (f = f_Q)$, unde $f_Q = \sum_{i=1}^{s} Q_i$ și mulțimea $\mathbf{Q} = \{Q_i \mid x_i \in X\}$ are proprietatea de minimalitate.*

Demonstrație. Presupunem $X = 1$; X fiind implicant al lui E conține pentru fiecare E_j fie variabila x_j, fie una oarecare dintre mulțimile $\mathbf{A}_j^k$, $k \in N_j^k$, adică fie $P_j \in \mathbf{Q}$, fie $\mathbf{R}_j^k \in \mathbf{Q}$. Dacă $P_j \in \mathbf{Q}$, atunci evident $P_j \Rightarrow f_Q$. Dacă $\mathbf{R}_j^k \in \mathbf{Q}$, atunci, ținînd seama de proprietatea $\mathscr{C}(\mathbf{R}_j^k) = P_j$, rezultă că și în acest caz $P_j \Rightarrow f_Q$ și deci $X = 1 \Rightarrow (P_j \Rightarrow f_Q)$ oricare ar fi $j \in N_j$. Rezultă implicația $f \Rightarrow f_Q$. Dar deoarece implicația inversă $f_Q \Rightarrow f$ este evidentă, rezultă egalitatea $f = f_Q$, ceea ce demonstrează prima parte a lemei.

Pentru terminarea demonstrației vom arăta că f_Q este expresie minimală a lui f. Presupunem contrariul. Atunci se poate elimina din $\mathbf{Q}$ cel puțin un termen, fie Q_i, astfel încît expresia rămasă prin înlăturarea lui Q_i din f să rămînă în continuare o expresie a lui f. Fie $\mathbf{Q}'$ mulțimea rămasă din $\mathbf{Q}$ prin eliminarea lui Q_i și f' expresia corespunzătoare. Dacă $f' = f_Q$, atunci în $\mathbf{Q}$ există o submulțime $\mathbf{R}_i = \{Q_{i1}, Q_{i2}, \ldots, Q_{ir_i}\}$ cu proprietatea $\mathscr{C}(\mathbf{R}_i) = Q_i$. Dar deoarece $x_{ik} \in \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_s\}$, $(\forall)\, k = 1, 2, \ldots, r_i$, rezultă că $X' = x_1 x_2 \ldots x_{j-1} x_{i+1} \ldots x_s$ este un implicant al lui E, contrar presupunerii că X este implicant prim al lui E. Deci $\mathbf{Q}$ are proprietatea de minimalitate, ceea ce demonstrează lema.

6.4.4. *Descrierea algoritmului de realizare minimală*

1. Se aduce funcția f la forma caracteristică f^c, de reuniune a tuturor implicanților săi primi. Fie $\mathbf{P}$ mulțimea implicanților primi ai lui f.

2. Se aplică algoritmul de determinare a consensurilor la funcția caracteristică sub următoarea formă puțin modificată :

Fie $P_1, P_2, \ldots, P_n$ termenii expresiei caracteristice f^c.

2.1. Se determină variabilele biforme. Fie acestea $x_1, x_2, \ldots, x_p$. Se face $x = x_1$.

2.2. Se determină consensurile în raport cu x. Printre aceste consensuri se vor găsi termeni din f^c sau consensuri noi. Fiecare consens în raport cu x astfel determinat se marchează cu indicii termenilor din f^c.

Dacă de exemplu $P_k = \mathscr{C}(P_i, P_j)$, atunci marcajul lui P_k va fi perechea (i, j). În general, dacă $P_k = \mathscr{C}(P_{i_1}, P_{i_2}, \ldots, P_{i_k})$, atunci marcajul va fi $(i_1, i_2, \ldots, i_k)$. La fel se procedează cu consensurile noi.

Ținînd seama că f^c conține toți implicanții primi ai funcției f, consensurile noi vor fi implicanți ai termenilor din f^c, dar nu vor fi înlăturați, întrucît folosesc la marcarea consensurilor în pașii următori.

2.3. Se adaugă consensurile noi la termenii din f^c.

2.4. Dacă $x = x_p$, se trece la pasul 3, altfel se face $x = x_{i+1}$ și se trece la pasul 2.2.

3. Se asociază termenilor expresiei f^c variabile booleene. Fie y_i variabila asociată termenului P_i, $i = 1, 2, \ldots, n$. Marcajului $(i_1, i_2, \ldots, i_k)$ i se asociază produsul $S_i = y_{i1} y_{i2} \ldots y_{ik}$.

4. Se construiesc expresiile $E_i = y_i + \sum_{j=1}^{n_i} S_i^j$, unde S_i^j corespunde marcajului j asociat implicantului prim P_i, în ipoteza că există mai multe marcaje iar n_i este numărul acestor marcaje.

5. Notăm prin $E_i^{(1)}$, $i = 1, 2, \ldots, n$, expresia obținută în pasul 4.

6. Se construiesc expresiile $E_i^{(2)}$, $i = 1, 2, \ldots, n$, din $E_i^{(1)}$ prin substituire după următoarea regulă. Dacă în $E_j^{(1)}$ apare y_k, atunci în $E_j^{(1)}$ se înlocuiește y_k prin $E_k^{(1)}$, pentru orice k pentru care $y_k \in E_j^{(1)}$. Expresia lui $E_j^{(1)}$ obținută după aceste substituții se va nota prin $E_j^{(2)}$. La fel se procedează cu toate expresiile $E_i^{(1)}$, $i = 1, 2, \ldots, n$.

7. În general, se obțin expresiile $E_i^{(k+1)}$ din $E_i^{(k)}$ la fel cum s-au obținut expresiile $E_i^{(2)}$ din $E_i^{(1)}$. Procesul încetează atunci cînd $E_i^{(k)} = E_i^{(k+1)}$ pentru orice i.

8. Se construiește expresia $E = \prod_{i=1}^{n} E_i$ și se aduce la forma caracteristică.

Implicanții de lungime ai lui E corespund expresiilor minimale astfel: dacă $y_1 y_2 \ldots y_p$ este un implicant prim de lungime minimă al lui E, atunci expresia $f_1 = \sum_{i=1}^{p} P_i$ este expresia minimă a lui f, unde P_i corespund variabilelor y_i, $i = 1, 2, \ldots, p$. În general pot exista mai multe expresii minime și un număr oarecare de expresii reduse.

Exemplul 1. $f^c = x_1x_3 + x_2x_3 + x_3\bar{x}_4 + x_1x_4 + x_2x_4 + \bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2\bar{x}_4$.

Notăm $P_1 = x_1x_3$, $P_2 = x_2x_3$, $P_3 = x_3\bar{x}_4$, $P_4 = x_1x_4$, $P_5 = x_2x_4$, $P_6 = \bar{x}_3x_4$, $P_7 = \bar{x}_1\bar{x}_2\bar{x}_3$, $P_8 = \bar{x}_1\bar{x}_2\bar{x}_4$.

Variabilele biforme sînt x_1, x_2, x_3, x_4.

1. Consensurile în raport cu x_1 sînt date de produsul

$$(x_3 + x_4)(\bar{x}_2\bar{x}_3 + \bar{x}_2\bar{x}_4) = \bar{x}_2x_3\bar{x}_4 + \bar{x}_2\bar{x}_3x_4.$$

Se adaugă consensurile $P_9 = \bar{x}_2x_3\bar{x}_4$ și $P_{10} = \bar{x}_2\bar{x}_3x_4$ cu marcajele (1, 8) și respectiv (4, 7).

2. Consensurile în raport cu x_2 sînt date de produsul

$$(x_3 + x_4)(\bar{x}_1\bar{x}_3 + \bar{x}_1\bar{x}_4 + x_3\bar{x}_4 + \bar{x}_3x_4) = x_3\bar{x}_4 + \bar{x}_3x_4.$$

Consensurile $P_3 = x_3\bar{x}_4$ și $P_6 = \bar{x}_3x_4$ se marchează prin (2,9)⇒ (1, 2, 8) și respectiv (5, 10) ⇒(4, 5, 7).

3. Consensurile în raport cu x_3 sînt date de produsul

$$(x_1 + x_2 + \bar{x}_4)(x_4 + \bar{x}_1\bar{x}_2) = x_1x_4 + x_2x_4 + \bar{x}_1\bar{x}_2\bar{x}_4.$$

Consensurile $P_4 = x_1x_4$, $P_5 = x_2x_4$ și P_8 primesc marcajele (1, 6), (2,6) și respectiv (3.7).

4. Consensurile în raport cu x_4 sînt date de produsul

$$(x_1 + x_2 + \bar{x}_3)(x_3 + \bar{x}_1\bar{x}_2) = x_1x_3 + x_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3.$$

Consensurile $P_1 = x_1x_3$, $P_2 = x_2x_3$ și P_7 primesc marcajele (3, 4), (3, 5) și respectiv (6, 8).

Expresiile E_i asociate implicanților primi P_i, $i = 1, 2, \ldots, 8$, sînt

$$E_1 = y_1 + y_3y_4,\ E_2 = y_2 + y_3y_5,\ E_3 = y_3 + y_1y_2y_8,\ E_4 = y_4 + y_1y_6,$$

$$E_5 = y_5 + y_2y_6,\ E_6 = y_6 + y_4y_5y_7,\ E_7 = y_7 + y_6y_8,\ E_8 = y_8 + y_3y_7.$$

Întrucît nu se pot face substituiri, se scrie E:

$$E = \prod_{i=1}^{8} E_i = (y_1 + y_3y_4)(y_2 + y_3y_5)(y_3 + y_1y_2y_8)(y_4 + y_1y_6) \times$$

$$\times (y_5 + y_2y_6)(y_6 + y_4y_5y_7)(y_7 + y_6y_8)(y_8 + y_3y_7) =$$

$$= y_1y_2y_3y_8 + y_3y_4y_5y_7 + y_1y_3y_5y_6y_7 + y_1y_3y_5y_6y_8 + y_1y_2y_4y_5y_7y_8.$$

Există deci două expresii minimale:

$$f_1 = x_1x_3 + x_2x_3 + \bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_4,$$

$$f_2 = x_3\bar{x}_4 + x_1x_4 + x_2x_4 + \bar{x}_1\bar{x}_2\bar{x}_3$$

și alte trei expresii reduse:

$$f_3 = x_1x_3 + x_3\bar{x}_4 + x_2x_4 + x_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3,$$

$$f_4 = x_1x_3 + x_3\bar{x}_4 + x_2x_4 + \bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_4,$$

$$f_5 = x_1x_3 + x_2x_3 + x_1x_4 + x_2x_4 + \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2\bar{x}_4.$$

Exemplul 2. $f^c = x_1x_5 + x_2x_6 + x_3\bar{x}_7 + x_4\bar{x}_5\bar{x}_6x_7 + x_1x_4\bar{x}_6x_7 + x_2x_4\bar{x}_5x_7 + x_1x_2x_4x_7 + x_3x_4\bar{x}_5\bar{x}_6 + x_1x_3x_4\bar{x}_6 + x_2x_3x_4\bar{x}_5 + x_1x_2x_3x_4$.

Notăm $P_1 = x_1x_5$, $P_2 = x_2x_6$, $P_3 = x_3\bar{x}_7$, $P_4 = x_4\bar{x}_5\bar{x}_6x_7$, $P_5 = x_1x_4\bar{x}_6x_7$, $P_6 = x_2x_4\bar{x}_5x_7$, $P_7 = x_1x_2x_4x_7$, $P_8 = x_3x_4\bar{x}_5\bar{x}_6$, $P_9 = x_1x_3x_4\bar{x}_6$, $P_{10} = x_2x_3x_4\bar{x}_5$, $P_{11} = x_1x_2x_3x_4$.

Variabilele biforme sînt x_5, x_6 și x_7.

Consensurile în raport cu x_5 sînt date de produsul

$$x_1(x_4\bar{x}_6x_7 + x_2x_4x_7 + x_3x_4\bar{x}_6 + x_2x_3x_4) = x_1x_4\bar{x}_6x_7 + x_1x_2x_4x_7 + x_1x_3x_4\bar{x}_6 + x_1x_2x_3x_4.$$

Consensurile P_5, P_7, P_9 și P_{11} se marchează prin (1,4), (1, 6), (1, 8) și respectiv (1, 10).

Consensurile în raport cu x_6 sînt date de produsul

$$x_2(x_4\bar{x}_5x_7 + x_1x_4x_7 + x_3x_4\bar{x}_5 + x_1x_3x_4) = x_2x_4\bar{x}_5x_7 + x_1x_2x_4x_7 + x_2x_3x_4\bar{x}_5 + x_1x_2x_3x_4.$$

Consensurile P_6, P_7, P_{10} și P_{11} se marchează prin (2, 4), (2, 5), (2, 8) și respectiv (2, 9).

Consensurile în raport cu x_7 sînt date de produsul

$$x_3(x_4\bar{x}_5\bar{x}_6 + x_1x_4\bar{x}_6 + x_2x_4\bar{x}_5 + x_1x_2x_4) = x_3x_4\bar{x}_5\bar{x} + x_1x_3x_4\bar{x} + x_2x_3x_4\bar{x}_5 + x_1x_2x_3x_4.$$

Consensurile P_8, P_9, P_{10} și P_{11} se marchează prin (3, 4), (3, 5), (3, 6) și respectiv (3, 7).

Consensurile P_1, P_2, P_3 și P_4 rămîn nemarcate.

Expresiile E_i asociate implicanților primi sînt:

$$E_1 = y_1,\ E_2 = y_2,\ E_3 = y_3,\ E_4 = y_4,\ E_5 = y_5 + y_1y_4,\ E_6 = y_6 + y_2y_4.$$

$$E_7 = y_7 + y_1y_6 + y_2y_5,\ E_8 = y_8 + y_3y_4,\ E_9 = y_9 + y_1y_8 + y_3y_5.$$

$$E_{10} = y_{10} + y_2y_8 + y_3y_6,\ E_{11} = y_{11} + y_1y_{10} + y_2y_{10} + y_3y_7.$$

După substituții expresiile E_i, $i = 1, \ldots, 11$, devin:

$$E_1 = y_1,\ E_2 = y_2,\ E_3 = y_3,\ E_4 = y_4.$$

$$E_5 = y_5 + y_1y_4,\ E_6 = y_6 + y_2y_4,\ E_7 = y_7 + y_1y_6 + y_2y_5 + y_1y_2y_4,$$

$$E_8 = y_8 + y_3y_4,\ E_9 = y_9 + y_1y_8 + y_3y_5 + y_1y_3y_4,\ E_{10} = y_{10} + y_2y_8 + y_3y_6 + y_2y_3y_4,\ E_{11} = y_{11} + y_1y_{10} + y_2y_9 + y_3y_7 + y_1y_2y_8 + y_1y_3y_6 + y_1y_2y_3y_4 + y_2y_3y_5 + y_1y_3y_6.$$

Se verifică imediat că $E = \prod_{i=1}^{11} E_i = y_1y_2y_3y_4$ și deci există o singură expresie minimă $f_1 = x_1x_5 + x_2x_6 + x_3x_7 + x_4x_5x_6x_7$.

Exemplele precedente demonstrează că metoda consensurilor și celelalte metode de simplificare sînt denumite astfel impropriu. În realitate, ele sînt metode de determinare a mulțimii implicanților primi ai unei funcții date și deci a expresiei caracteristice a acestei funcții.

În acest capitol am prezentat o formă îmbunătățită a metodei consensurilor și un algoritm nou de obținere a expresiilor minimale.

6.5. Metoda celui mai mare divizor comun

6.5.1. *Introducere*

Proprietatea de maximalitate a hiperfețelor asociate implicanților primi și relația de divizibilitate introdusă în mulțimea implicanților unității ne permit să punem problema determinării mulțimii implicanților primi ai unei funcții booleene date într-un mod diferit de cel din 6.3. În loc de a obține implicanții primi prin compunerea succesivă a vecinilor se poate merge pe calea determinării directe a hiperfețelor maximale. Un astfel de algoritm va fi prezentat în paragraful următor [31].

Fie $\mathbf{C} = \{c_1, c_2, \ldots, c_a\}$ mulțimea numerelor convenționale asociate constituenților unității ai unei funcții booleene f. O hiperfață (a, b) din $\mathbf{B}_2^n$ este inclusă în f, dacă $x \in \mathbf{C}$ pentru orice $a \leqslant x \leqslant b$.

Lema 1. *Unui implicant prim (a, b) al lui f îi corespunde o hiperfață maximală în f.*

Demonstrație. Fie $X = (a, b)$ un implicant prim al lui f și $Y = (c, d)$ un implicant al lui f cu proprietatea $Y \Rightarrow X$, $Y \neq X$.

Din $Y \Rightarrow X$ rezultă $(c, d) \leqslant (a, b)$ și conform teoremei din 4.3.3, $(d, c+d) \subset (b, a+b)$, adică hiperfața corespunzătoare lui X include hiperfața corespunzătoare lui Y, ceea ce demonstrează lema.

Rezultatele precedente ne permit să dăm un algoritm pentru simplificarea funcțiilor booleene bazat pe căutarea hiperfețelor maximale incluse în funcția f dată.

Descrierea algoritmului. 1. Se scrie f prin numerele convenționale sau vectorii asociați constituenților unității.

2. Se ia un număr convențional m, începînd cu cel mai mare și se înscrie într-un tabel $\mathbf{T}_0$ divizorii acestuia incluși în f.

3. Se caută vecinii lui m incluși în $\mathbf{T}_0$ Fie aceștia m_1, $m_2, \ldots, m_k$. Se înscriu într-un tabel $\mathbf{T}_1$ hiperfețele (m_i, m), $i = 1, 2, \ldots, k$.

4. Se află cei mai mari minoranți comuni $m_i \cap m_j$, $i, j = 1, 2, \ldots, k$, și se înscrie într-un tabel $\mathbf{T}_2$ hiperfețele $(m_i \cap m_j, m)$ pentru orice $m_i \cap m_j \in \mathbf{T}_0$, eliminîndu-se în același timp hiperfețele (m_i, m) și (m_j, m) din $\mathbf{T}_1$. Hiperfețele (m_r, m) care rămîn în $\mathbf{T}_1$ neeliminate sînt maximale și se înscriu într-un tabel $\mathbf{T}$ de implicanți primi.

Procesul continuă construind un tabel $\mathbf{T}_{i+1}$ din $\mathbf{T}_i$ la fel cum s-a obținut $\mathbf{T}_2$ din $\mathbf{T}_1$, pînă se ajunge la un tabel $\mathbf{T}_{i+1}$ care nu conține nici un element sau care conține un singur element care este o hiperfață maximală și se înscrie în tabelul de implicanți primi.

5. Se repetă operațiile 2—4 pentru toate numerele convenționale, verificînd la fiecare înscriere a hiperfețelor în tabelul de implicanți primi dacă acestea nu se includ în hiperfețele înscrise anterior.

Exemplu

$f = \{0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14\}$,

$n = 14$, $\mathbf{T}_0 = \{0, 2, 4, 6, 8, 10, 14\}$,

$\mathbf{T}_1 = \{(6, 14), (10, 14)\}$, $6 \cap 10 = 2$,

$\mathbf{T}_2 = \{(2, 14)\}$, $\mathbf{T} = \{(2, 14)\}$.

(2, 14) este hiperfață maximală și-i corespunde implicantul prim (12, 2).

$m = 11$, $\mathbf{T}_0 = \{0, 1, 2, 3, 8, 9, 10, 11\}$,

$\mathbf{T}_1 = \{(3, 11), (9, 11), (10, 11)\}$,

$\mathbf{T}_2 = \{(1, 11), (2, 11), (8, 11)\}$,

$\mathbf{T}_3 = \{(0, 11)\}$, $\mathbf{T} = \{(0, 11)\}$

și se obține implicantul prim (11, 0).

$m = 10$, $\mathbf{T}_0 = \{0, 2, 8, 10\}$,

$\mathbf{T}_1 = \{(2, 10), (8, 10)\}$,

$\mathbf{T}_2 = \{(0, 10)\}$,

$(0, 10) \subset (0, 11)$.

Dealtfel, hiperfețele care corespund oricărui divizor al lui 11 se includ în (0, 11), de aceea mai departe nu vom mai repeta procesul pentru aceștia:

$m = 7$, $\mathbf{T}_0 = \{0, 1, 2, 3, 4, 6, 7\}$.

$\mathbf{T}_1 = \{(3, 7), (6, 7)\}$,

$\mathbf{T}_2 = \{(2, 6)\}$, $\mathbf{T} = \{(2, 7)\}$

și se obține implicantul (5, 2).

$m = 6$, $\mathbf{T}_0 = \{0, 2, 4, 6\}$,

$\mathbf{T}_1 = \{(2, 6), (4, 6)\}$,

$\mathbf{T}_2 = \{(0, 6)\}$, $\mathbf{T} = \{(0, 6)\}$

și se obține implicantul prim (6, 0).

Implicanții primi ai funcției f sînt:

$(12, 2) = (1100, 0010)$, $(5, 2) = (0101, 0010)$,

$(11, 0) = (1011, 0000)$, $(6, 0) = (0110, 0000)$.

Notînd cele patru variabile cu x, y, z, u, aceștia se vor scrie $z\bar{u}$, $\bar{y}$, $\bar{x}z$, $\bar{x}\bar{u}$ care conduc la expresia

$$f = \bar{y} + \bar{x}z + \bar{x}\bar{u} + z\bar{u}.$$

Deoarece f nu conține variabile biforme, această expresie a funcției f este minimală.

CAPITOLUL 7

SIMPLIFICAREA FUNCȚIILOR DE VARIABILE POLIVALENTE

7.1. Generalități

Teoria funcțiilor de variabile polivalente nu este mult mai complicată decît teoria funcțiilor de variabile bivalente. În capitolele precedente s-au extins în mod natural metodele algebrei booleene la funcții de variabile polivalente. Cu unele restricții inerente au fost extinse metodele de reprezentare a funcțiilor booleene, formele normale ale acestora și metodele de rezolvare a ecațiilor booleene la funcții de variabile polivalente. Fără îndoială că toate metodele clasice, înțelegînd prin acestea metodele algebrei booleene, își găsesc corespondent în teoria funcțiilor de variabile polivalente. În acest capitol vom prezenta extinderile metodelor de simplificare a funțciilor booleene la funcțiile de variabile polivalente.

7.2. Metoda Quine-McCluskey generalizată

7.2.1. *Algoritmul de determinare a implicanților primi*

Metoda Quine-McCluskey poate fi extinsă cu ușurință la cazul polivalent. În acest ultim caz posibilitățile de simplificare se bazează pe egalitatea $ax^0 + ax^1 + \ldots + ax^{k-1} = a$, adevărată pentru orice variabilă x, k-valentă. Prin a am notat un produs de variabile polivalente.

Definiția 1. Doi implicanți a și b ai unei funcții de variabile polivalente care diferă numai prin forma unei variabile se numesc *cvasivecini*. Dacă vectorul $a \oplus b$ are toate componentele nule, cu excepția uneia care este egală cu 1, atunci implicanții a și b se numesc *vecini*. Operația de determinare a implicanților primi dată și de relația $ax^0 + ax^1 + \ldots + ax^{k-1} = a$ se numește operația de *compunere a vecinilor*.

Fie $\mathbf{M} = \{M_1, M_2, \ldots, M_m\}$ mulțimea constituenților unității ai funcției $f(x_1, x_2, \ldots, x_n)$. Fiecărui constituent al unității $M_i = x_1^{a_{i1}} x_2^{a_{i2}} \ldots x_n^{a_{in}}$ îi asociem vectorul $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ și numărul convențional $P_i = \sum_{i=1}^{n} a_i w_i$, unde $w_i = k_{i+1} w_{i+1}$, $i = 1, 2, \ldots, n-1$, $w_n = 1$.

Descrierea algoritmului. 1. Se aduce funcția la forma de reuniune a constituenților unității $f = \sum_{i=1}^{m} M_i$, $M_i = x_1^{a_{i1}} \ldots x_n^{a_{in}}$. Fie $\mathbf{M} = \{M_1, M_2 \ldots, M_n\}$ mulțimea constituenților unității ai funcției f.

2. Fiecărui constituent al unității $M_i = x_1^{a_{i1}} \ldots x_n^{a_{in}}$ i se asociază vectorul $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ și numărul convențional $p_i = \sum_{i=1}^{n} a_i w_i$, $w_i = k_{i+1} w_{i+1}$, $w_n = 1$, iar k_i, $i = 1, 2, \ldots, n$, sînt k-valențele variabilelor x_i. Vectorii $\mathbf{a}_i$ se așază într-un tabel $\mathbf{T}_0$ în ordinea crescătoare a numerelor convenționale.

3. Se aplică operația de compunere a vecinilor pentru toate mulțimile de k implicanți cvasivecini doi cîte doi în raport cu aceeași variabilă k-valentă.

4. Se marchează locurile variabilelor care dispar prin compunere, iar vectorului rezultant i se asociază mulțimea numerelor convenționale ale vectorilor care intră în compunere. Rezultatul se trece într-un nou tabel $\mathbf{T}_1$.

5. Vectorii din $\mathbf{T}_0$ care nu intră în nici una dintre operațiile de compunere din pasul 3 sînt implicanți primi și se trec într-un tabel **T** de implicanți primi.

6. Dacă $\mathbf{T}_1$ este vid, procesul încetează. Dacă nu, se șterg din $\mathbf{T}_0$ vectorii care trec în **T**, se trec vectorii din $\mathbf{T}_1$ în $\mathbf{T}_0$ și se șterg din $\mathbf{T}_1$, apoi se trece la pasul 3.

Exemplu. Să se determine implicanții primi pentru funcția $f(a,b,x)$, dată de mulțimea de numere convenționale {4, 5, 9, 11, 12, 13, 14, 15, 16, 20, 22, 23, 26}.

Tabelul $\mathbf{T}_0$ al vectorilor asociați constituenților unității este dat în tabelul 7.1. Rezultatul compunerii vecinilor este trecut în tabelul 7.2 în

Tabelul 7.1

p	a	b	x	p	a	b	x
9	1	0	0	13	1	1	1
12	1	1	0	14	1	1	2
4	0	1	1	22	2	1	1
15	1	2	0	16	1	2	1
5	0	1	2	23	2	1	2
11	1	0	2	26	1	2	2
20	2	1	0				

care în prima coloană s-au trecut numerele convenționale ale vectorilor care intră în compunere. Toți vectorii din tabelul 7.1 au intrat în compunere. Aplicînd pasul 3 tabelului 7.2, se constată că nu se mai pot face compuneri.

Tabelul 7.2

	a	b	x
9—12—15	1	—	0
12—13—14	1	1	—
4—13—22	—	1	1
5—14—23	—	1	1
11—14—26	1	—	2
15—16—26	1	2	—
20—22—23	2	1	—

Rezultă că toți vectorii din acest tabel corespund implicanților primi. Expresia caracteristică a funcției f este următoarea:

$$f(a, b, x) = a^1 x^0 + a^1 b^1 + b^1 x^1 + b^1 x^2 + a^1 x^2 + a^1 b^2 + a^2 b^1.$$

7.2.2. *Algoritmul de determinare a expresiilor minimale*

Pentru găsirea expresiilor minimale se folosește algoritmul descris în 6.2.2. Se fac notațiile

$M_1 = a^1b^0x^0$, $M_2 = a^1b^1x^0$, $M_3 = a^0b^1x^1$, $M_4 = a^1b^2x^0$,
$M_5 = a^0b^1x^2$, $M_6 = a^1b^0x^2$, $M_7 = a^2b^1x^0$, $M_8 = a^1b^1x^1$,
$M_9 = a^1b^1x^2$, $M_{10} = a^2b^1x^1$, $M_{11} = a^1b^2x^1$, $M_{12} = a^2b^1x^2$,
$M_{13} = a^1b^2x^2$ și $P_1 = a^1x^0$, $P_2 = a^1b^1$, $P_3 = b^1x^1$,
$P_4 = b^1x^2$, $P_5 = a^1x^2$, $P_6 = a^1b^2$, $P_7 = a^2b^1$;

fiecărui constituent al unității M_i i se asociază vectorul $\mathbf{Q}_i = (Q_{i1}, Q_{i2}, \ldots, Q_{i7})$, unde $Q_{ik} = 1$ dacă $M_i \Rightarrow P_k$ și $Q_{ik} = 0$ în caz contrar. Rezultatul se trece în tabelul 7.3.

Tabelul 7.3

	P_1	P_2	P_3	P_4	P_5	P_6	P_7
M_1	1	0	0	0	0	0	0
M_2	1	1	0	0	0	0	0
M_3	0	0	1	0	0	0	0
M_4	1	0	0	0	0	1	0
M_5	0	0	0	1	0	0	0
M_6	0	0	0	0	1	0	0
M_7	0	0	0	0	0	0	1
M_8	0	1	1	0	0	0	0
M_9	0	0	0	1	1	0	0
M_{10}	0	0	1	0	0	0	1
M_{11}	0	0	0	0	0	1	0
M_{12}	0	0	1	0	0	0	1
M_{13}	0	0	0	0	1	1	0

Se asociază funcția

$$F(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = \prod_{i=1}^{13} \sum_{k=1}^{7} a_{ik} P_k =$$

$$= P_1(P_1+P_2)P_3(P_1+P_6)P_4P_5P_7(P_2+P_3)(P_4+P_5)(P_3+P_7)\times$$
$$\times P_6(P_4+P_7)(P_5+P_6) = P_1P_3P_4P_5P_6P_7$$ care conduce la expresia minimală a funcției $f(a, b, x) = a^1x^0 + b^1x^1 + b^1x^2 + a^1x^2 + a^1b^2 + a^2b^1$. Aceasta diferă de expresia caracteristică numai prin lipsa implicantului prim a^1b^1.

7.3. Metoda hiperfețelor maximale

Vom generaliza metoda din 6.3 la funcții de variabile polivalente. Algoritmul dat evită înregistrările multiple în procesul de simplificare și este recomandat în cazul utilizării calculatorului electronic. Acest algoritm [31] utilizează rezultatele descrise în 4.3.

Fie $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ vectorul n dimensional asociat constituentului unității $X = x_1^{p_1} x_2^{p_2} \ldots x_n^{p_n}$, unde vectorul $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ este k-valent. Dacă vectorul $\mathbf{p}$ este divizor al vectorului $\mathbf{q}$, între acești vectori există relația $\mathbf{p} + (\mathbf{p} \oplus \mathbf{q}) = \mathbf{q}$, notată și $\mathbf{p} \leqslant \mathbf{q}$. După cum am arătat în cap. 4, condiția necesară și suficientă pentru ca să existe relația de divizibilitate $\mathbf{p} + (\mathbf{p} \oplus \mathbf{q}) = \mathbf{q}$ între vectorii $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ și $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ este ca $p_i \leqslant q_i$, $(\forall)\ i = 1, 2, \ldots, n$.

Dacă $d_i = \min(p_i, q_i)$ și $s_i = \max\limits_i (p_i, q_i)$, $i = 1, 2, \ldots, n$, atunci vectorul $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ este cel mai mare divizor comun, iar vectorul $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ este cel mai mic multiplu comun al vectorilor $\mathbf{p}$ și $\mathbf{q}$.

Mulțimea vectorilor asociați constituenților unității din funcțiile de variabile polivalente formează latice cu ajutorul relației de divizibilitate (v. cap. 4), iar mulțimea divizorilor unui vector arbitrar din $\mathbf{B}_k^n$ formează pseudogrup cu operația $\oplus$. Dacă $\mathbf{H} = (\mathbf{p}, \mathbf{q})$ este o cvasifață în $\mathbf{B}_k^n$, atunci ea este complet determinată de primul său element p și de ultimul element q și pentru orice $\mathbf{r} \in \mathbf{H}$ există relația $\mathbf{p} \leqslant \mathbf{r} \leqslant \mathbf{q}$.

Înainte de a trece la descrierea algoritmului vom face unele completări față de cele demonstrate în cap. 4.

Convenim să asociem pseudogrupului $\mathbf{B}_k^n$ produsul

$$P = (x_1^0 + x_1^1 + \ldots + x_1^{k_1-1})(x_2^0 + x_2^1 + \ldots + x_2^{k_2-1}) \ldots \ldots (x_n^0 + x_n^1 + \ldots + x_n^{k_n-1}) = 1.$$

Lema 1. *Dacă* $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ *și* $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ *sînt doi vectori k-valenți, între care există relația* $\mathbf{p} \leqslant \mathbf{q}$, *atunci cvasifeței* $\mathbf{H} = (\mathbf{p}, \mathbf{q})$ *îi corespunde produsul*

$$P_H = (x_1^{p_1} + x_1^{p_1+1} + \ldots x_1^{q_1})(x_2^{p_2} + x_2^{p_2+1} + \ldots + x_2^{q_2}) \ldots \ldots (x_n^{p_n} + x_n^{p_n+1} + \ldots + x_n^{q_n}).$$

Demonstrație. Din $\mathbf{p} \leqslant \mathbf{q}$ se obține $p_i \leqslant q_i$, $(\forall)\ i = 1, 2, \ldots, n$, și ținînd seama de definiția lui $\mathbf{H}$, orice $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ cu proprietatea $\tau \in \mathbf{H}$ îndeplinește condiția $p_i \leqslant r_i \leqslant q_i$ pentru orice i, de unde rezultă că

$$(x_1^{r_1} x_2^{r_2} \ldots x_n^{r_n} = 1) \Rightarrow (P_H = 1).$$

Convenim ca în produsul P_H să înlăturăm termenii pentru care $p_i = 0$, $q_i = k_i - 1$, care sînt egali cu 1. Dacă notăm $\mathbf{I}_H$ mulțimea de indici ai variabilelor x_i care apar efectiv în P_H și $S_i = \sum\limits_{j \in [p_i,\ q_i]} x_j^i$, atunci P_H se scrie sub forma $P_H = \prod\limits_{i \in \mathbf{I}_H} S_i$. Dacă $\mathbf{p} = \mathbf{q} = \mathbf{r}$, cvasifața $\mathbf{H}$ se reduce la vectorul $\mathbf{r}$ iar P_H se reduce la produsul de variabile $P_r = \prod\limits_{i \in \mathbf{I}_r} x_i^{r_i}$, unde $\mathbf{I}$ este mulțimea de indici ai variabilelor care apar efectiv în P_r.

Definiția 1. Spunem că produsul P_H asociat hiperfeței H este *implicant prim generalizat* al unei funcții f de variabile polivalente, dacă este implicant al funcției f și dacă implicantul unității P'_H obținut din P_H prin înlăturarea cel puțin a unui termen S_i încetează de a mai fi implicant al lui f.

Întrucît nu se nasc confuzii, în loc de implicant prim generalizat vom spune simplu implicant prim.

Pentru $\mathbf{H} = \mathbf{r}$ noțiunile de implicant și de implicant prim se reduc la noțiunile obișnuite.

Definiția 2. *O cvasifață* $\mathbf{H}$ este *maximală* în f dacă pentru orice cvasifață $\mathbf{G}$ cu proprietățile $\mathbf{H} \subset \mathbf{G}$, $P_H \leqslant P_G$, $(P_G = 1) \Rightarrow (f = 1)$ rezultă $\mathbf{H} = \mathbf{G}$, $P_H = P_G$.

Lema 2. *Dacă o cvasifață* $\mathbf{H}$ *este maximală în* f, *atunci implicantul* P_H *corespunzător este un implicant prim al lui* f.

Demonstrație. Presupunem contrariul. Fie P_G implicantul obținut din P_H prin înlăturarea unui termen oarecare S_i. Dar hiperfața $\mathbf{G} = (\mathbf{u}, \mathbf{v})$ asociată lui P_G are componenta $u_i = 0$, $v_i = k_i - 1$ și toate celelalte componente egale cu componentele lui $\mathbf{H}$. Dar $0 \leqslant p_i,\ q_i \leqslant k_i - 1$ și cel puțin una dintre inegalități este strictă. Rezultă că $\mathbf{H} \subset \mathbf{G}$, $\mathbf{H} \neq \mathbf{G}$ contrar presupunerii că $\mathbf{H}$ este maximală în f.

O consecință a acestei leme este că determinarea implicanților primi se reduce la căutarea cvasifețelor maximale în f.

Dacă $\mathbf{H} = (\mathbf{p}, \mathbf{q})$, atunci, conform celor arătate în 4.5.3, implicantul P_H asociat se scrie sub forma $P_H = (\mathbf{p} \oplus \mathbf{q}, \mathbf{p})$, unde $p_i \oplus q_i = q_i - p_i$. Această ultimă relație este justificată de faptul că $q_i \geqslant p_i$, $(\forall)\ i = 1, 2, \ldots, n$.

Observație. $q_i - p_i + 1$ reprezintă numărul de termeni în suma S_i iar $x_i^{p_i}$ este primul termen din suma S_i, oricare ar fi $i \in \mathbf{I}_H$. Se poate spune deci că $\mathbf{p} \oplus \mathbf{q}$ indică numărul de termeni care apar în sumele S_i, $i = 1, 2, \ldots, n$, iar $\mathbf{p}$ primul termen al fiecărei sume.

Între doi implicanți (a, b) și (c, d) ai unei funcții f se introduce operația $\oplus$ dată de relația $(a, b) \oplus (c, d) = (a \oplus c, b \oplus d)$.

Definiția 3. Doi *implicanți* (a, b) și (c, d) se numesc *vecini*, dacă $(a, b) \oplus (c, d) = (\theta, Q_i)$, unde Q_i este un vector care are toate componentele nule, cu excepția componentei i care este egală cu 1 iar θ este vectorul nul. Rezultă că doi implicanți sînt vecini, dacă $a = c$ și dacă b și d sînt la rîndul lor vecini.

Operația de compunere a vecinilor o vom nota ca și pentru cazul binar cu semnul $\boxplus$ și vom defini astfel: $(a, b) \boxplus (a, c) = (a + (b \oplus c),\ d)$, unde $d = \min(b, c)$.

Lema 3. *Dacă (a, b) și (a, c) sînt implicanți obținuți într-un pas oarecare al simplificării și sînt vecini, atunci expresia $(a + (b \oplus c), d)$ reprezintă implicantul obținut prin compunerea lui (a, b) cu (a, c).*

Demonstrație. Dacă b și c sînt vecini, atunci $b \oplus c = Q_i$ și $a + (b \oplus c)$ se obține din a prin adăugarea lui 1 la componenta i, restul de componente rămînînd neschimbate.

Lema 4. *Între implicanții* $(\mathbf{a}, \mathbf{b})$, $(\mathbf{a}, \mathbf{c})$, $(\mathbf{a} + \mathbf{b} \oplus \mathbf{c}), \mathbf{d})$ *există relațiile* $(\mathbf{a}, \mathbf{b}) \leqslant (\mathbf{a} + (\mathbf{b} \oplus \mathbf{c}), \mathbf{d})$, $(\mathbf{a}, \mathbf{c}) \leqslant (\mathbf{a} + (\mathbf{b} \oplus \mathbf{c}), \mathbf{d})$.

Demonstrație. Să arătăm mai întîi că $(\mathbf{a}, \mathbf{b}) \leqslant (\mathbf{a} + (\mathbf{b} \oplus \mathbf{c}), \mathbf{d})$. Într-adevăr, $a_i \leqslant a_i + (b_i \oplus c_i)$ pentru orice $i = 1, 2, \ldots, n$ și $\mathbf{d} = \min(\mathbf{b}, \mathbf{c}) \leqslant b$. Fie $\mathscr{P}_a = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$

un constituent al unității în funcția polivalentă de n variabile $f(x_1, x_2, \ldots, x_n)$. Convenim să asociem variabilelor $x_1, x_2, \ldots, x_n$ ponderi, după cum urmează: lui x_1 îi asociem ponderea 1; apoi, dacă x_i are ponderea w_i, lui x_{i+1} i se asociază ponderea $w_{i+1} = k_i w_i$, unde k_i este k-valența variabilei x_i iar constituentului unității $\mathscr{P}_a = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ i se asociază numărul convențional $Q_a = \sum_{i=1}^{n} a_i w_i$. În general, unui vector $(a_1, a_2, \ldots, a_n)$ i se asociază numărul convențional Q_a dat de relația precedentă.

Definiția 3. Spunem că vectorul $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ este *mai mare* decît vectorul $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ și scriem $\mathbf{a} > \mathbf{b}$ dacă $Q_a > Q_b$.

Sîntem în măsură acum să demonstrăm teorema care generalizează teorema din 6.3.2 și care ne permite să extindem algoritmul din 6.3.3 la funcțiile de variabile polivalente.

Presupunem că implicantul $(\mathbf{A}, \mathbf{B})$ al unei funcții oarecare de variabile polivalente se obține în mai multe feluri într-un pas oarecare al procesului de sinteză prin compunerea vecinilor. Presupunem că aceste compuneri sînt următoarele:

$$(\mathbf{A}_1, \mathbf{B}) \boxplus (\mathbf{A}_1, \mathbf{B}_1) = (\mathbf{A}, \mathbf{B}),\ (\mathbf{A}_2, \mathbf{B}) \boxplus$$

$$\boxplus (\mathbf{A}_2, \mathbf{B}_2) = (\mathbf{A}, \mathbf{B}), \ldots, (\mathbf{A}_r, \mathbf{B}) \oplus (\mathbf{A}_r, \mathbf{B}_r) = (\mathbf{A}, \mathbf{B}),$$

unde $\mathbf{A}_i = (a_{i1}, \ldots, a_{in})$, $\mathbf{B}_i = (b_{i1}, \ldots, b_{in})$, $(\forall)\ i \in [1, r]$.

$$\mathbf{A} = (a_1, \ldots, a_n),\ \mathbf{B} = (b_1, \ldots, b_n).$$

Fie $\mathbf{E}_i = \mathbf{B} \oplus \mathbf{B}_i$. Vectorul $\mathbf{E}_i$ are toate componentele egale cu 0 cu excepția uneia care este egală cu 1.

Fără a scădea din generalitatea problemei se poate presupune că $\mathbf{E}_i$ are componenta $e_{ii} = 1$ și $e_{ij} = 0$ pentru $i \neq j$.

Teorema 1. *Dacă* $(\mathbf{A}_i, \mathbf{B}) \boxplus (\mathbf{A}_i, \mathbf{B}_i) = (\mathbf{A}, \mathbf{B})$, $i = 1, 2, \ldots, r$, *sînt toate modurile de obținere a implicantului* $(\mathbf{A}, \mathbf{B})$ *prin compunerea vecinilor, atunci există un indice s, și numai unul, astfel încît* $(a_{ss} + 1)\mathbf{E}_s > \mathbf{A}_s$.

Demonstrație. Conform presupunerii, $\mathbf{A}_i$ se poate scrie sub forma $\mathbf{A}_i = \mathbf{A}'_i + a_{ii}\,\mathbf{E}_i$, unde $\mathbf{A}'_i$ are toate componentele egale cu ale lui $\mathbf{A}_i$, cu excepția componentei i care este egală cu 0. Apoi $\mathbf{E}_i \neq \mathbf{E}_j$, $i \neq j$, $(\forall)\ i,\ j \in \{1, 2, \ldots, r\}$. Într-adevăr, în caz contrar din $\mathbf{E}_i = \mathbf{E}_j$ rezultă $\mathbf{B} \oplus \mathbf{B}_i = \mathbf{B} \oplus \mathbf{B}_j$, adică $\mathbf{B}_i = \mathbf{B}_j$ și din $\mathbf{A}_i + (\mathbf{B} \oplus \mathbf{B}_i) = \mathbf{A}_j + (\mathbf{B} \oplus \mathbf{B}_j)$ rezultă $\mathbf{A}_i = \mathbf{A}_j$, adică $(\mathbf{A}_i, \mathbf{B}_i) = (\mathbf{A}_i, \mathbf{B}_j)$ contrar presupunerii. Mai departe rezultă $\mathbf{A}_i + (\mathbf{B} \oplus \mathbf{B}_i) = \mathbf{A}'_i + (a_{ii} + 1)\,\mathbf{E}_i = \mathbf{A}$, $(\forall)\ i \in \{1, 2, \ldots, r\}$. Dar componenta a_i a lui $\mathbf{A}$ este tocmai $a_{ii} + 1$ pentru orice $i \in \{1, 2, \ldots, r\}$.

Să arătăm că pentru orice $k \notin \{1,2, \ldots, r\}$, $a_k = 0$. Presupunem contrariul. Atunci există $a_k \neq 0$, $k \notin \{1,2, \ldots, r\}$. Fie $\mathbf{E}$ vectorul cu componenta $e_k = 1$ și restul de componente nule, $\mathbf{C}$ vectorul care are componentele egale cu ale lui $\mathbf{A}$ cu excepția componentei a_k, egală cu $a_k - 1$ și $\mathbf{D}$ vectorul cu componentele egale cu componentele lui $\mathbf{B}$, cu excepția componentei b_k, egală cu $b_k + 1$. Este evident că $(\mathbf{C}, \mathbf{B}) \boxplus (\mathbf{C}, \mathbf{D}) = (\mathbf{A}, \mathbf{B})$ cu $\mathbf{B} \oplus \mathbf{D} = \mathbf{E}$, $\mathbf{B}$ și $\mathbf{D}$ vecini. Dar relația precedentă nu se găsește printre relațiile de compunere din enunțul teoremei, contrar presupunerii, de unde rezultă că ipoteza $a_k \neq 0$, $k \notin \{1, 2, \ldots, r\}$, este falsă. Se poate scrie $\mathbf{A} = \sum_{i \in \{1,2,\ldots,r\}} (a_{ii} + 1)\,\mathbf{E}_i$, de unde rezultă

$a_i = a_{ii} + 1$, $(\forall)\ i \in \{1, 2, \ldots, r\}$, $a_i = 0$ în caz contrar.

Apoi din $\mathbf{A}_i + (\mathbf{B} \oplus \mathbf{B}_i) = \mathbf{A}_j + \mathbf{E}_j = \mathbf{A}$ rezultă imediat $\mathbf{A}_j = \sum_{i \neq j} (a_{ii} + 1)\mathbf{E}_i + a_{jj}\mathbf{E}_j$, $(\forall)\ j \in \{1, 2, \ldots, r\}$, sau, ceea ce este același lucru, $\mathbf{A}_j = \sum_{i \neq j} a_i\,\mathbf{E}'_i + (a_j - 1)\,\mathbf{E}_j$, $(\forall)$ $j \in \{1, 2, \ldots, r\}$.

Să arătăm că există un indice s și unul singur cu proprietatea $(a_{ss} + 1)\mathbf{E}_s > \mathbf{A}_s$. Fie $P_1, P_2, \ldots, P_r$ numerele convenționale asociate vectorilor $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_r$. Fără a pierde din generalitate le putem aranja astfel încît $\mathbf{E}_1$ să aibă componenta 1 în coloana 1 și în general $\mathbf{E}_i$ să aibă componenta 1 în coloana i, $(\forall)\ i \in \{1, 2, \ldots, r\}$. Întrucît $\mathbf{E}_i \neq \mathbf{E}_j$ pentru $i \neq j$, vom avea de asemenea $P_i \neq P_j$. Există deci o relație de ordine strictă între $P_1, P_2, \ldots, P_r$. Ținînd seama de relația $P_{i+1} = k_i P_i$, $k_i \geqslant 2$, $P_1 = 1$,

obținem $P_1 < P_2 < \ldots < P_r$. Mai mult $\sum_{k=1}^{i} P_k < P_{i+1}$, $(\forall) i \in \{1, 2, \ldots, r\}$. Într-adevăr, $1 < k_1(k_2 - 1) \Rightarrow 1 + k_1 < k_1 k_2$, $k_1 \geqslant 2\ k_2 \geqslant 2$, de unde rezultă că inegalitatea $\sum_{k=1}^{i} P_k < P_{i+1}$ se verifică pentru $i = 1$. Presupunem că inegalitatea se verifică pentru $i - 1$. Atunci se obține succesiv $\sum_{k=1}^{i-1} P_k < P_i \Rightarrow \sum_{k=1}^{i} P_k < 2P_i \leqslant P_{i+1}$, $k_i \geqslant 2$, adică se verifică în general. Deoarece $a_{ii} \leqslant k_i - 1$, rezultă $\sum_{i=1}^{r-1} (a_{ir} + 1)P_i \leqslant \sum_{i=1}^{r-1} k_i P_i < P_r$ și deci $\sum_{i=1}^{r-1} (a_{ii} + 1)\,P_i + a_{rr} P_r < (a_{rr} + 1)P_r$, adică $\sum_{i=1}^{r-1} (a_{ii} + 1)\mathbf{E}_i + a_{rr}\,\mathbf{E}_r < (a_{rr} + 1)\,\mathbf{E}_r$ sau $\mathbf{A}_r < (a_{rr} + 1)\mathbf{E}_r$.

Pe de altă parte, inegalitatea $P_j < P_r$, $(\forall) j < r$, implică $P_j < \sum_{i \neq j} (a_{ii} + 1)P_i \Rightarrow (a_{jj} + 1)P_j < \sum_{i \neq j} (a_{ii} + 1)P_i + a_{jj}\,P_j \Rightarrow (a_{jj} + 1)\,\mathbf{E}_j < \mathbf{A}_j$, $(\forall)$ $j < r$. Deci se poate lua $s = r$ în condițiile teoremei.

Cu aceasta se poate trece la descrierea algoritmului care repetă aproape cuvînt cu cuvînt algoritmul din cazul bivalent.

Descrierea algoritmului. Fie $f(x_1, x_2, \ldots, x_n)$ o funcție de variabile polivalente. Fiecărui constituent al unității $x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ îi corespunde cvasifața $(\mathbf{a}, \mathbf{a})$, unde $\mathbf{a}$ este vectorul $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ și respectiv implicantul unității $(\mathbf{a} \oplus \mathbf{a}, \mathbf{a}) = (\mathbf{0}, \mathbf{a})$, unde $\mathbf{0}$ este vectorul nul.

1. Se înscriu într-un tabel $\mathbf{M}_0$ toți implicanții $(\mathbf{0}, \mathbf{a})$, corespunzători constituenților unității ai funcției

$$f(x_1, x_2, \ldots, x_n).$$

2. Se construiește un șir de mulțimi de perechi $\mathbf{M}_0\mathbf{M}_1, \ldots, \mathbf{M}_k, \ldots$, trecerea de la mulțimea $\mathbf{M}_k$ la mulțimea $\mathbf{M}_{k+1}$ făcîndu-se astfel :

2.1. Pentru $(\mathbf{a}, \mathbf{b}) \in \mathbf{M}_k$ și $(\mathbf{c}, \mathbf{d}) \in \mathbf{M}_k$, oricare ar fi perechile $(\mathbf{a}, \mathbf{b})$, $(\mathbf{c}, \mathbf{d})$, se aplică operația $\oplus$.

2.2. Dacă rezultatul operației $\oplus$ este de forma $(\mathbf{0}, \mathbf{E}_i)$, adică $\mathbf{a} = \mathbf{c}$ și $\mathbf{b}$ vecin cu $\mathbf{d}$ și dacă $\mathbf{b} < \mathbf{d}$, se verifică ine-

galitatea $\mathbf{a} < (a_i + 1)\,\mathbf{E}_i$, i fiind componenta egală cu 1 în $\mathbf{E}_i$ iar a_i componenta i a lui $\mathbf{a}$.

2.3. Dacă inegalitatea precedentă este îndeplinită, se aplică perechilor $(\mathbf{a}, \mathbf{b})$, $(\mathbf{a}, \mathbf{d})$ operația $\boxplus$ de compunere a vecinilor și rezultatul se înscrie în $\mathbf{M}_{k+1}$.

2.4. O pereche $(\mathbf{a}, \mathbf{b}) \in \mathbf{M}_k$ pentru care nu există nici o pereche $(\mathbf{c}, \mathbf{d})$ din $\mathbf{M}_k$ astfel încît $(\mathbf{a}, \mathbf{b}) \oplus (\mathbf{c}, \mathbf{d}) = (\boldsymbol{\theta}, \mathbf{E}_i)$ este un implicant prim și se înscrie într-un tabel de implicanți primi.

2.5. Procesul încetează cînd $\mathbf{M}_{k+1}$ are cel mult un element care este implicant prim. Dacă $\mathbf{M}_{k+1}$ nu îndeplinește această condiție, se repetă pașii 2.1 la 2.4 pentru $\mathbf{M}_{k+1}$.

În locul vectorilor se pot utiliza numerele convenționale asociate și operațiile $\oplus$ și $\boxplus$ cu numere convenționale.

Exemplu. Să se simplifice funcția

$$f(a, b, c) = a^1b^0c^0 + a^1b^1c^0 + a^1b^2c^0 + a^1b^1c^1 + a^1b^2c^1 + a^1b^0c^2 + a^1b^1c^2 + a^1b^2c^2.$$

1. Tabelul este

$\mathbf{M}_0 = \{[(0, 0, 0), (1, 0, 0)], [(0, 0, 0), (1, 1, 0)], [(0, 0, 0), (1, 2, 0)], [(0, 0, 0), (1, 0, 1)], [(0, 0, 0), (1, 1, 1)], [(0, 0, 0), (1, 2, 1)], [(0, 0, 0), (1, 0, 2)], [(0, 0, 0), (1, 1, 2)], [(0, 0, 0), (1, 2, 2)]\}$.

2.1. Perechile de perechi vecine sînt următoarele:

$\{[(0, 0, 0), (1, 0, 0)], [(0, 0, 0), (1, 1, 0)]\}$,

$\{[(0, 0, 0), (1, 0, 0)], [(0, 0, 0), (1, 0, 1)]\}$,

$\{[(0, 0, 0), (1, 1, 0)], [(0, 0, 0), (1, 2, 0)]\}$,

$\{[(0, 0, 0), (1, 1, 0)], [(0, 0, 0), (1, 1, 1)]\}$,

$\{[(0, 0, 0), (1, 2, 0)], [(0, 0, 0), (1, 2, 1)]\}$,

$\{[(0, 0, 0), (1, 0, 1)], [(0, 0, 0), (1, 1, 1)]\}$,

$\{[(0, 0, 0), (1, 0, 1)], [(0, 0, 0), (1, 0, 2)]\}$,

$\{[(0, 0, 0), (1, 1, 1)], [(0, 0, 0), (1, 2, 1)]\}$,

$\{[(0, 0, 0), (1, 1, 1)], [(1, 0, 0), (1, 1, 2)]\}$,

$\{[(0, 0, 0), (1, 2, 1)], [(0, 0, 0), (1, 2, 2)]\}$,

$\{[(0, 0, 0), (1, 0, 2)], [(0, 0, 0), (1, 1, 2)]\}$,

$\{[(0, 0, 0), (1, 1, 2)], [(0, 0, 0), (1, 2, 2)]\}$.

2.2. În toate cazurile $(a_{ii} + 1)\,\mathbf{E}_i > \mathbf{A}_i$. Se trece deci la punctul 2.3.

2.3. Se aplică perechilor de la punctul 2.1 operația $\boxplus$ și rezultatul se trece în $\mathbf{M}_1$. Tabelul va fi

$\mathbf{M}_1 = \{[(0, 1, 0), (1, 0, 0)], [(0, 0, 1), (1, 0, 0)], [(0, 1, 0), (1, 1, 0)], [(0, 0, 1), (1, 1, 0)], [(0, 0, 1), (1, 2, 0)], [(0, 1, 0), (1, 0, 1)], [(0, 0, 1), (1, 0, 1)], [(0, 1, 0), (1, 1, 1)], [(0, 0, 1), (1, 1, 1)], [(0, 0, 1), (1, 2, 1)], [(0, 1, 0), (1, 0, 2)], [(0, 1, 0), (1, 1, 2)]\}$.

2.4. Toate perechile din $\mathbf{M}_0$ apar în tabelul de la punctul 2.1. Se trece deci la punctul 2.5.

2.5. Tabelul $\mathbf{M}_1$ nu este vid, deci se trece la punctul 2.1 și se repetă procesul pentru $\mathbf{M}_1$.

2.1. Perechile de perechi vecine ale lui $\mathbf{M}_1$ sînt următoarele:

$\{[(0, 1, 0), (1, 0, 0),], [(0, 1, 0), (1, 1, 0)]\}$,

$\{[(0, 1, 0), (1, 0, 0)], [(0, 1, 0), (1, 0, 1)]\}$,

$\{[(0, 0, 1), (1, 0, 0)], [(0, 0, 1), (1, 1, 0)]\}$,

$\{[(0, 0, 1), (1, 0, 0)], [(0, 0, 1), (1, 0, 1)]\}$,

$\{[(0, 1, 0), (1, 1, 0)], [(0, 1, 0), (1, 1, 1)]\}$,

$\{[(0, 0, 1), (1, 1, 0)], [(0, 0, 1), (1, 1, 1)]\}$,

$\{[(0, 0, 1), (1, 1, 0)], [(0, 0, 1), (1, 2, 0)]\}$,

$\{[(0, 0, 1), (1, 2, 0)], [(0, 0, 1), (1, 2, 1)]\}$,

$\{[(0, 1, 0), (1, 0, 1)], [(0, 1, 0), (1, 1, 1)]\}$,

$\{[(0, 1, 0), (1, 0, 1)], [(0, 1, 0), (1, 0, 2)]\}$,

$\{[(0, 0, 1), (1, 0, 1)], [(0, 0, 1), (1, 1, 1)]\}$,

$\{[(0, 1, 0), (1, 1, 1)], [(0, 1, 0), (1, 1, 2)]\}$,

$\{[(0, 0, 1), (1, 1, 1)], [(0, 0, 1), (1, 2, 1)]\}$,

$\{[(0, 1, 0), (1, 0, 2)], [(0, 1, 0), (1, 1, 2)]\}$.

2.2. Perechile pentru care se verifică inegalitatea $(a_{ii} + 1)\mathbf{E}_i > \mathbf{A}_i$ sînt următoarele:

$\{[(0, 1, 0), (1, 0, 0)], [(0, 1, 0), (1, 1, 0)]\}$,

$\{[(0, 0, 1), (1, 0, 0)], [(0, 0, 1), (1, 1, 0)]\}$,

$\{[(0, 0, 1), (1, 0, 0)], [(0, 0, 1), (1, 0, 1)]\}$,

$\{[(0, 0, 1), (1, 1, 0)], [(0, 0, 1), (1, 1, 1)]\}$,

{[(0, 0, 1), (1, 2, 0)], [(0, 0, 1), (1, 2, 1)]},

{[(0, 1, 0), (1, 0, 1)], [(0, 1, 0), (1, 1, 1)]},

{[(0, 0, 1), (1, 0, 1), [(0, 0, 1), (1, 1, 1)]},

{[(0, 0, 1), (1, 1, 1)], [(0, 0, 1), (1, 2, 1)},

{[(0, 1, 0), (1, 0, 2), [(0, 1, 0), (1, 1, 2)]},

{[(0, 0, 1), (1, 1, 0)], [(0, 0, 1), (1, 2, 0)]}.

2.3. Se aplică perechilor de la punctul 2.2 operația $\oplus$ și rezultatul se înscrie în $\mathbf{M}_2$. Tabelul va fi

$\mathbf{M}_2 = \{[(0, 2, 0), (1, 0, 0)], [(0, 1, 1), (1, 0, 0)], [(0, 0, 2), (1, 0, 0)], [(0, 0, 2), (1, 1, 0)], [(0, 0, 2), (1, 2, 0)], [(0, 2, 0), (1, 0, 1)], [(0, 1, 1), (1, 0, 1)], [(0, 1, 1), (1, 1, 1)], [(0, 2, 0), (1, 0, 2)], [(0, 1, 1), (1, 1, 0)]\}$.

2.4. Toate perechile din $\mathbf{M}_1$ apar în tabelul de la punctul 2.1. Se trece deci la punctul 2.5.

2.5. $\mathbf{M}_2$ nu este vid, deci se trece la 2.1 și se repetă procesul pentru $\mathbf{M}_2$.

2.1. Perechile de perechi vecine ale lui $\mathbf{M}_2$ sînt următoarele :

{[(0, 2, 0), (1, 0, 0)], [(0, 2, 0), (1, 0, 1)]},

{[(0, 1, 1), (1, 0, 0)], [(0, 1, 1), (1, 0, 1)]},

{[(0, 0, 2), (1, 0, 0)], [(0, 0, 2), (1, 1, 0)]},

{[(0, 0, 2), (1, 1, 0)], [(0, 0, 2), (1, 2, 0)]},

{[(0, 2, 0), (1, 0, 1)], [(0, 2, 0), (1, 0, 2)]},

{[(0, 1, 1), (1, 0, 1)], [(0, 1, 1), (1, 1, 1)]},

{[(0, 1, 1), (1, 0, 0)], [(0, 1, 1), (1, 1, 0)]}.

{[(0, 1, 1), (1, 1, 0)], [(0, 1, 1), (1, 1, 1)]}.

2.2. Perechile pentru care se verifică inegalitatea $(a_{ii} \div 1)\mathbf{E}_i > \mathbf{A}_i$ sînt următoarele :

{[(0, 0, 2), (1, 0, 0)], [(0, 0, 2), (1, 1, 0)]},

{[(0, 0, 2), (1, 1, 0)], [(0, 0, 2), (1, 2, 0)]},

{[(0, 1, 1), (1, 0, 1)], [(0, 1, 1), (1, 1, 1)]},

{[(0, 1, 1), (1, 0, 0)], [(0, 1, 1), (1, 1, 0)]}.

2.3. Se aplică perechilor de la punctul 2.2 operația $\pm$ și rezultatul se înscrie în $\mathbf{M}_3$. Tabelul va fi

$\mathbf{M}_3 = \{[(0, 1, 2), (1, 0, 0)], [(0, 2, 1), (1, 0, 0)],$
$[(0, 1, 2), (1, 1, 0)], [(0, 2, 1), (1, 0, 1)]\}$.

2.4. Toate perechile din $\mathbf{M}_2$ apar în tabelul de la punctul 2.1. Se trece la punctul 2.5.

2.5. Tabelul $\mathbf{M}_3$ nu este vid, deci se trece la punctul 2.1 și se repetă procesul pentru $\mathbf{M}_3$.

2.1. Perechile de perechi vecine ale lui $\mathbf{M}_3$ sînt următoarele :

{[(0, 1, 2), (1, 0, 0)], [(0, 1, 2), (1, 1, 0)]},

{[(0, 2, 1), (1, 0, 0)], [(0, 2, 1), (1, 0, 1)]}.

2.2. Inegalitatea $(a_{ii} + 1)\ \mathbf{E}_i > \mathbf{A}_i$ se verifică pentru prima dintre perechile precedente.

2.3. Se aplică operația $\boxplus$ la această pereche și se obține tabelul $\mathbf{M}_4 = [(0, 2, 2), (1, 0, 0)]$.

2.4. Toate perechile din $\mathbf{M}_3$ sînt cuprinse în 2.1.

2.5. Tabelul $\mathbf{M}_4$ are un singur element care este singurul implicant prim al funcției date. Acesta se scrie

$$a^1(b^0 + b^1 + b^2)(c^0 + c^1 + c^2) = a^1,$$

rezu · care se poate verifica direct.

Algoritmul descris prezintă avantaje în cazul programării simplificării pe calculatorul electronic, întrucît permite alegerea implicanților fără repetări.

7.4. Metoda consensurilor generalizată

Noțiunea de consens a fost generalizată în 6.4, de la consensul a doi termeni într-o funcție booleană la consensul a n termeni într-o astfel de funcție. Vom da o nouă generalizare a acestei noțiuni, extinzînd-o la funcții de variabile polivalente, care ne va permite să dăm o metodă de simplificare a funcțiilor de variabile polivalente care generalizează metoda lui Tison și metoda Quine McCluskey din cazul bivalent.

Metoda consensurilor generalizată se bazează pe implicația $a_0a_1 \ldots a_{k-1} \Rightarrow a_0x^0 + a_1x^1 + \ldots + a_{k-1}x^{k-1}$, denumită implicația consensului, în care x^i, $i = 0, 1, 2, \ldots, k-1$, este o variabilă k-valentă iar $a_0, a_1 \ldots, a_{k-1}$ sînt produse de astfel de variabile. Evident pentru ca implicația consensului să nu fie banală este necesar ca produsul $a_0a_1 \ldots a_{k-1}$ să fie nenul, adică $a_0, a_1, \ldots, a_{k-1}$ să conțină

numai variabile monoforme. Implicația consensului se verifică ușor, observînd că

$$a_0a_1 \ldots a_{k-1} = 1 \Rightarrow (a_i = 1,\ i = 0, 1, 2, \ldots, k-1) \Rightarrow$$
$$\Rightarrow a_0x^0 + a_1x^1 + \ldots + a_{k-1}x^{k-1} = x^0 + x^1 + \ldots$$
$$\ldots + x^{k-1} = 1.$$

Observație. Pentru $a_0 = a_1 = \ldots = a_{k-1} = a$ implicația se reduce la egalitate, ceea ce indică sensul în care metoda consensurilor generalizează metoda Quine-McCluskey.

Ca și în cazul bivalent se poate vorbi de consensul a n termeni. Toate proprietățile consensurilor date în 6.4 se regăsesc la funcțiile de variabile polivalente.

Fie $n+1$ termeni $b_0, b_1, b_2, \ldots, b_n$, cu proprietățile $b_0 \Rightarrow$ $\Rightarrow \sum_{i=1}^{n}{}'b_i, b_0 \neq \sum_{i=1}^{n}{}'b_i$, unde $\sum_{i=1}^{n}{}'b_i$ se obține din reuniunea termenilor $b_1, b_2, \ldots, b_n$ prin eliminarea cel puțin a unui termen.

Definiție. Termenul b_0 care îndeplinește condițiile precedente se numește *consensul termenilor* $b_1, b_2, \ldots, b_n$.

Orice variabilă monoformă din mulțimea termenilor $b_1, b_2, \ldots, b_n$ este inclusă în b_0 și, reciproc, b_0 conține numai variabilele monoforme din mulțimea acestor termeni.

Fie $f(x_1, x_2, \ldots, x_n)$ o funcție de variabile polivalente și x_i o variabilă k-valentă care apare efectiv în $f(x_1, x_2, \ldots, x_n)$.

Lema 1. *Dacă x_i este k-formă, adică dacă în $f(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ apar efectiv $x_i^0, x_i^1, \ldots, x_i^{k-1}$, atunci există implicația*

$$\prod_{a=0}^{k-1} f(x_1, x_2, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) \Rightarrow f(x_1, x_2, \ldots, x_n).$$

Demonstrație. Într-adevăr, în acest caz se poate scrie

$$f(x_1, x_2, \ldots, x_n) = \sum_{a=0}^{k-1} f(x_1, x_2, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n)x_i^a.$$

Dar $P_i = \prod_{a=0}^{k-1} f(x_1, x_2, \ldots, x_{i-1}, a, x_{i+1} \ldots, x_n) = 1$ implică $f(x_1, x_2, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) = 1$, $(\forall)a = 0, 1, 2, \ldots, k-1$. Dacă se înlocuiesc termenii $f(x_1, x_2, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n)$ în expresia precedentă a lui $f(x_1, x_2, \ldots, x_n)$, rezultă

$$f(x_1, x_2, \ldots, x_n) = \sum_{a=0}^{k-1} x_i^{a_i} = 1.$$

Evident, pentru existența consensului este necesar și suficient ca x_i să fie k-formă în $f(x_1, x_2, \ldots, x_n)$.

Consecință. *În general, din $P_i \Rightarrow f(x_1, x_2, \ldots, x_n)$ și din $A \Rightarrow P_i$ rezultă $A \Rightarrow f(x_1, x_2, \ldots, x_n)$, unde*

$$P_i = \prod_{a=0}^{k-1} f(x_1 \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n).$$

Această consecință ne conduce la o nouă metodă de simplificare a funcțiilor de variabile polivalente, care generalizează metoda consensurilor din cazul bivalent. Sîntem în măsură acum să dăm algoritmul respectiv.

Descrierea algoritmului. 1. Se determină variabilele k-forme, k fiind valențele variabilelor respective, și se aranjează într-o ordine arbitrară: $x_1, x_2, \ldots, x_n$.

2. Se stabilește tabelul reziduurilor în raport cu x_1^a, $f(a, x_2, \ldots, x_n)$, $a = 0, 1, 2, \ldots, k_1 - 1$ (se iau în considerare termenii care conțin efectiv pe x_1^a).

3. Se face produsul $P_1 = \prod_{a=0}^{k_1-1} f(a, x_2, \ldots, x_n)$.

Implicanții lui P_1 diferiți de 0 constituie consensuri în raport cu x_1.

4. Se construiește funcția $f_1 = P_1 + f$ și se elimină termenii redondanți.

5. Se repetă pașii 2 — 4 pentru restul de variabile din mulțimea $x_1, x_2, \ldots, x_r$ și se construiește f_{i+1} din f_i la fel cum s-a construit f_1 din f.

6. Procesul încetează după aplicarea pașilor 2—4 la toate variabilele k-forme, $x_1, x_2, \ldots, x_r$.

Exemplu. Să se simplifice funcția :

$$f(a, b, x) = a^1b^0x^0 + a^1b^1x^0 + a^0b^1x^1 + a^1b^1x^1 + a^1b^1x^2 + a^1b^2x^0 + a^2b^1x^1 + a^0b^1x^2 + a^0b^2x^1 + a^0b^2x^2 + a^1b^0x^2 + a^1b^2x^1 + a^1b^2x^2 + a^2b^0x^0 + a^2b^0x^2 + a^2b^1x^0 + a^2b^1x^2 + a^2b^2x^0 + a^2b^2x^1 + a^2b^2x^2.$$

1. Mulțimea variabilelor triforme este a, b, x.
2. Reziduurile în raport cu a^u, $u = 0, 1, 2$, sînt $f(0, b, x) = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2$, $f(1, b, x) = b^0x^0 + b^1x^0 + b^1x^1 + b^1x^2 + b^2x^0 + b^0x^2 + b^2x^1 + b^2x^2$, $f(2, b, x) = b^1x^1 + b^0x^0 + b^0x^2 + b^1x^0 + b^1x^2 + b^2x^0 + b^2x^1 + b^2x^2$.

3. $P_1 = (b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2)(b^0x^0 + b^1x^0 + b^1x^1 + b^1x^2 + b^2x^0 + b^0x^2 + b^2x^1 + b^2x^2)(b^1x^1 + b^0x^0 + b^0x^2 + b^1x^0 + b^1x^2 + b^2x^0 + b^2x^1 + b^2x^2) = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2$.

4. $f_1 = P_1 + f = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2 + a^1b^0x^0 + a^1b^1x^0 + a^1b^2x^0 + a^1b^0x^2 + a^2b^0x^0 + a^2b^0x^2 + a^2b^1x^0 + a^2b^2x^0$.

Observație. Din f s-au scris numai termenii neredondanți. Se trece la aplicarea pașilor 2—4 variabilei b și funcției f_1.

2^1. Reziduurile în raport cu b^u, $u = 0, 1, 2$, sînt

$$f_1(a, 0, x) = a^1x^0 + a^1x^2 + a^2x^0 + a^2x^2,$$

$f_1(a, 1, x) = x^1 + x^2 + a^1x^0 + a^2x^0$, $f_1(a, 2, x) = x^1 + x^2 + a^1x^0 + a^2x^0$

3^1. $P_2 = (a^1x^0 + a^1x^2 + a^2x^0 + a^2x^2)(x^1 + x^2 + a^1x^0 + a^2x^0) \times (x^1 + x^2 + a^1x^0 + a^2x^0) = a^1x^0 + a^2x^0 + a^1x^2 + a^2x^2$.

3^1. $f_2 = P_2 + f_1 = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2 + a^1x^0 + a^2x^0 + a^1x^2 + a^2x^2$.

Se aplică pașii 2—4 funcției f_2.

2^2. Reziduurile în raport cu x^u, $u = 0, 1, 2$, sînt

$$f_2(a, b, 0) = a^1 + a^2, \quad f_2(a, b, 1) = b^1 + b^2,$$
$$f_2(a, b, 2) = b^1 + b^2 + a^1 + a^2.$$

3^2. $P_3 = (a^1 + a^2)(b^1 + b^2)(b^1 + b^2 + a^1 + a^2) = a^1b^1 + a^1b^2 + a^2b^1 + a^2b^2$.

4^2. $f_3 = P_3 + f_2 = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2 + a^1x^0 + a^2x^0 + a^1x^2 + a^2x^2 + a^1b^1 + a^1b^2 + a^2b^1 + a^2b^2$.

Pașii 2—4 fiind aplicați la toate variabilele a, b, x, f_3 reprezintă expresia simplificată a funcției date $f(a, b, x)$.

Determinarea expresiilor minimale ale funcției $f(a, b, x)$. Aceste expresii se obțin, ca și în cazul bivalent (vezi 6.4), folosind algoritmul :

1. Se aduce funcția la forma caracteristică (reuniunea tuturor implicanților primi).
2. Pentru fiecare implicant prim P_j se determină toate submulțimile de implicanți primi $\mathbf{R}_j^k$ care au proprietatea că P_j este consensul elementelor din $\mathbf{R}_j^k$.
3. Se asociază implicanților primi P_j variabile booleene y_j și se construiește funcția asociată $E = \prod_{j=1}^{r} E_j$, $E_j = y_j + \sum_k S_j^k$; S_j^k fiind produsul variabilelor asociate elementeor din $\mathbf{R}_j^k$, E_j se construiește ca și în cazul bivalent.
4. Se aduce funcția E la forma caracteristică E^c.
5. Se consideră implicanții primi ai lui E cu număr minim de litere și pentru fiecare astfel de implicant $y_1, y_2, \ldots, y_s$ se scrie expresia $\sum_{i=1}^{s} P_i$, care este o realizare minimală a lui f.

În continuare vom determina expresiile minimale pentru funcția din exemplul precedent.

1. Expresia caracteristică a funcției $f(a, b, x)$ este

$$^c = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2 + a^1x^0 + a^2x^0 + a^1x^2 + a^2x^2 + a^1b^1 + a^2b^1 + a^2b^2 + a^1b^2.$$

2. Facem notațiile

$$P_1 = b^1x^1, \ P_2 = b^1x^2, \ P_3 = b^2x^1, \ P_4 = b^2x^2,$$
$$P_5 = a^1x^0, \ P_6 = a^2x^0, \ P_7 = a^1x^2, \ P_8 = a^2x^2, \ P_9 = a^1b^1,$$
$$P_{10} = a^1b^2, \ P_{11} = a^2b^1, \ P_{12} = a^2b^2.$$

Singura variabilă triformă este x. Implicanții în care apar x^0, x^1 și x^2 sînt : x^0 în P_5 și P_6, x^1 în P_1 și P_3, x^2 în P_2, P_4, P_7 și P_8.

Vom nota cu P_5', P_6', P_1', P_3', P_2', P_4', P_7' și P_8' resturile implicanților respectivi după eliminarea lui x^0, x^1 și respectiv x^2. Consensurile în raport cu x vor fi date de produsul

$$(P_5' + P_6')(P_1' + P_3')(P_2' + P_4' + P_7' + P_8') = P_1'P_2'P_5' + P_1'P_2'P_6' + P_1'P_5'P_7' + P_1'P_6'P_8' + P_3'P_4'P_5' + P_3'P_4'P_6' + P_3'P_5'P_7' + P_3'P_6'P_8',$$

unde $P_1' = P_2' = b^1$, $P_3' = P_4' = b^2$, $P_5' = P_7' = a^1$, $P_6' = P_8' = a^2$. Nu s-au scris termenii care se anulează.

După ce se înlocuiesc $P'_1, P'_2, \ldots, P'_8$ prin expresiile lor în raport cu a și b, rezultă

$$(a^1 + a^2)(b^1 + b^2)(a^1 + a^2 + b^1 + b^2) = a^1b^1 + a^1a^2 + a^2b^1 + a^2b^2 = P_9 + P_{10} + P_{11} + P_{12},$$

ceea ce conduce la implicațiile:

$$P_9 \Rightarrow P_1 + P_2 + P_5,\ P_9 \Rightarrow P_1 + P_5 + P_7,$$

$$P_{10} \Rightarrow P_3 + P_4 + P_5,\ P_{10} \Rightarrow P_3 + P_5 + P_7,\ P_{11} \Rightarrow P_1 + P_2 + P_6,$$

$$P_{11} \Rightarrow P_1 + P_6 + P_8,\ P_{12} \Rightarrow P_3 + P_4 + P_6,\ P_{12} \Rightarrow P_3 + P_6 + P_8.$$

3. Se construiesc iterativ expresiile E_j, ca în cazul bivalent. Primele valori E_j vor fi:

$$E^1_1 = y_1,\ E^1_2 = y_2,\ E^1_3 = y_3,\ E^1_4 = y_4,\ E^1_5 = y_5,\ E^1_6 = y_6,\ E^1_7 = y_7,$$

$$E^1_8 = y_8,\ E^1_9 = y_9 + y_1y_2y_5 + y_1y_5y_7,\ E^1_{10} = y_{10} + y_3y_4y_5 + y_3y_5y_7,$$

$$E^1_{11} = y_{11} + y_1y_2y_6 + y_1y_6y_8,\ E^1_{12} = y_{12} + y_3y_4y_6 + y_3y_6y_8.$$

Deoarece y_9, y_{10}, y_{11} și y_{12} nu apar în partea dreaptă decît în expresiile corespunzătoare E^1_9, E^1_{10}, E^1_{11}, E^1_{12}, procesul încetează la primul pas și deci avem $E_i = E^1_i$ pentru orice i și expresia E devine

$$E = \prod_{j=1}^{11} E_j =$$

$$= y_1y_2y_3y_4y_5y_6y_7y_8(y_9 + y_1y_2y_5 + y_1y_5y_7)(y_{10} + y_3y_4y_5 + y_3y_5y_7)(y_{11} + y_1y_2y_6 + y_1y_6y_8)(y_{12} + y_3y_4y_6 + y_3y_6y_8) = y_1y_2y_3y_4y_5y_6y_7y_8,$$

care conduce la expresia minimă

$f(a, b, x) = b^1x^1 + b^1x^2 + b^2x^1 + b^2x^2 + a^1x^0 + a^2x^0 + a^1x^2 + a^2x^2 = P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8$ care are numai opt termeni în timp ce expresia caracteristică are 12 termeni.

7.5. Metoda celui mai mare divizor comun

Această metodă este o transpunere aproape cuvînt cu cuvînt a metodei corespunzătoare din cazul bivalent. Vom utiliza noțiunea generalizată de implicant dată în 7.3.

Fie

$$P = (x_{i_1}^{p_1} + x_{i_1}^{p_1+1} + \ldots + x_{i_1}^{q_1})(x_{i_2}^{p_2} + x_{i_2}^{p_2+1} + \ldots + x_{i_2}^{q_2}) \ldots (x_{i_k}^{p_k} + x_{i_k}^{p_k+1} + \ldots + x_{i_k}^{q_k})$$

un astfel de implicant. Vom nota cu $\mathbf{p}$ vectorul care are componentele $i_1, i_2, \ldots, i_k$ egale respectiv cu $p_1, p_2, \ldots, p_k$, restul de componente fiind nule și cu $\mathbf{q}$ vectorul care are componentele $i_1, i_2, \ldots, i_k$ egale respectiv cu $q_1, q_2, \ldots, q_k$, restul de componente fiind egale cu 0. Evident, $\mathbf{p}$ și $\mathbf{q}$ caracterizează implicantul P. Odată dați $\mathbf{p}$ și $\mathbf{q}$, implicantul P va fi complet determinat.

În general, dacă $\mathbf{p}$ și $\mathbf{q}$ sînt doi vectori k-valenți, $\mathbf{p} \leqslant \mathbf{q}$, atunci $p_i \oplus q_i = q_i - p_i$. Fiecare sumă din P conține $q_i - p_i + 1$ termeni.

Cum s-a mai arătat, perechea $(\mathbf{p} \oplus \mathbf{q}, \mathbf{p})$ determină complet implicantul P. Componentele vectorului $\mathbf{p} \oplus \mathbf{q}$ indică numărul de termeni din fiecare sumă a expresiei lui P, numărul acestor termeni fiind egal cu $q_i - p_i + 1$ pentru orice i. Vectorul $\mathbf{p}$ indică cel mai mic termen din fiecare sumă.

Constituenții unității $P_a = x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ se scriu sub forma $(\mathbf{0}, \mathbf{a})$, unde $\mathbf{0}$ este vectorul nul iar $\mathbf{a} = (a_1, a_2, \ldots, a_n)$.

Metoda de simplificare descrisă se bazează pe proprietățile laticei codurilor k-valente din 4.3.4.

Relația de ordine între doi vectori $\mathbf{p}$ și $\mathbf{q}$ ai laticei codurilor k-valente $\mathbf{p} \leqslant \mathbf{q}$ este echivalentă cu următoarea relație: $\mathbf{p} + (\mathbf{p} \oplus \mathbf{q}) = \mathbf{q}$.

Vom reaminti noțiunile de cel mai mare divizor comun și de cel mai mic multiplu comun date în 4.3.4. Fie $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ și $\mathbf{d} = (d_1, d_2, \ldots, d_n)$, $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, unde $d_i = \min(q_i, p_i)$, $s_i = \max(q_i, p_i)$. Prin definiție, vectorul $\mathbf{d}$ se numește cel mai mare divizor comun iar vectorul $\mathbf{s}$ cel mai mic multiplu comun al vectorilor $\mathbf{p}$ și $\mathbf{q}$.

Am văzut în 7.3 că o cvasifață și respectiv o hiperfață sînt complet determinate de două elemente ale lor. Fie $\mathbf{H} = (\mathbf{p}, \mathbf{q})$ o astfel de cvasifață. Acestei cvasifețe îi

corespunde implicantul $(\mathbf{p} \oplus \mathbf{q}, \mathbf{p})$. Implicanții primi ai unei funcții f corespund cvasifețelor maximale în f. Folosind aceste noțiuni, vom putea trece la prezentarea algoritmului.

Descrierea algoritmului. 1. Se scrie funcția dată f sub formă de reuniune de constituenți ai unității. Se asociază fiecărui constituent al unității un vector $\mathbf{p}$ și se scrie sub formă de implicant $(\boldsymbol{\theta}, \mathbf{p})$, $\boldsymbol{\theta}$ fiind vectorul nul.

2. Se înscriu implicanții $(\boldsymbol{\theta}, \mathbf{p})$ într-un tabel $\mathbf{T}$.

3. Se împarte mulțimea $\mathbf{P} = \{(s_i, p_i) \mid i = 1, 2, \ldots, m\}$ a implicanților din $\mathbf{T}$ în submulțimi $\mathbf{P}^k$, cu proprietățile $\bigcup_k \mathbf{P}^k = \mathbf{P}$,

$$[(s_i, p_i) \in \mathbf{P}^k,\ (s_j, p_j) \in \mathbf{P}^k] \Rightarrow s_i = s_j,$$

$$[(s_i, p_i) \in \mathbf{P}^k, (s_j, p_j) \in \mathbf{P}^m,\ m \neq k] \Rightarrow s_i \neq s_j.$$

4. Se înscriu într-un tabel $\mathbf{T}_0$ toți vectorii $\mathbf{p}$ care apar în implicanții $(\mathbf{s}, \mathbf{p})$ ai unei submulțimi $\mathbf{P}^k$, ordonați după mărimea numerelor convenționale asociate.

5. Se ia fiecare vector $\mathbf{p}$ din $\mathbf{T}_0$ începînd cu cel mai mare și se caută vecinii lui $\mathbf{p}$ incluși în $\mathbf{T}_0$. Fie aceștia $\mathbf{p}_1$, $\mathbf{p}_2, \ldots, \mathbf{p}_k$ și se înscriu într-un tabel $\mathbf{T}_1$ cvasifețele $(\mathbf{p}_i, \mathbf{p})$, $i = 1, 2, \ldots, k$.

6. Se află cei mai mari divizori comuni $\mathbf{p}_i \cap \mathbf{p}_j$, $i, j = 1, 2, \ldots, k$, și se înscriu într-un tabel $\mathbf{T}_2$ cvasifețele $(\mathbf{p}_i \cap \mathbf{p}_j, \mathbf{p})$, pentru orice $\mathbf{p}_i \cap \mathbf{p}_j \in \mathbf{T}_0$, eliminîndu-se în același timp cvasifețele $(\mathbf{p}_i, \mathbf{p})$ și $(\mathbf{p}_j, \mathbf{p})$ din $\mathbf{T}_1$.

Cvasifețele $(\mathbf{p}_r, \mathbf{p})$ care rămîn în $\mathbf{T}_1$ neeliminate sînt maximale și se înscriu într-un tabel $\mathbf{T}_c$.

7. Procesul continuă construind un tabel $\mathbf{T}_{k+1}$ din $\mathbf{T}_k$ la fel cum am construit $\mathbf{T}_2$ din $\mathbf{T}_1$, pînă se ajunge la un tabel $\mathbf{T}_{k+1}$ care nu conține nici un element sau care conține un singur element, care este o cvasifață maximală și se înscrie în $\mathbf{T}_c$.

8. Se repetă pașii 5—7 pentru toți vectorii $\mathbf{p}$ din $\mathbf{T}_0$, avînd grijă de fiecare dată cînd se fac înscrieri în tabelul $\mathbf{T}_c$ să nu fie înscrise cvasifețele care se includ în cvasifețe deja înscrise.

9. Se repetă pașii 4—8 pentru toate submulțimile $\mathbf{P}^k$ din $\mathbf{P}$.

10. Fie $(\mathbf{p}_i, \mathbf{q}_i)$, $i = 1, 2, \ldots, r$, cvasifețele incluse în $\mathbf{T}_c$ după pasul al nouălea. Dacă $\mathbf{p}_i$ și $\mathbf{q}_i$ provin din implicanții $(\mathbf{s}, \mathbf{p}_i)$ și respectiv $(\mathbf{s}, \mathbf{q}_i)$, se calculează implicanții corespunzători cvasifeței $(\mathbf{p}_i, \mathbf{q}_i)$, cu formula $[\mathbf{s} + (\mathbf{p}_i \oplus \oplus \mathbf{q}_i), \mathbf{p}_i]$.

11. Fie $(\mathbf{s}_i, \mathbf{p}_i)$, $i = 1, 2, \ldots, r$, implicanții obținuți în pasul 10. Se elimină din $\mathbf{T}$ toți implicanții care se includ în acești implicanți. Implicanții neeliminați sînt primi și se înscriu în tabelul de implicanți primi $\mathbf{T}_p$, eliminîndu-se în același timp din $\mathbf{T}$.

12. Implicanții $(\mathbf{s}_i, \mathbf{p})$ obținuți în pasul 10 care se bucură de proprietatea că $\mathbf{s}_i \neq \mathbf{s}_j$, oricare ar fi $j \neq i$, $j = 1, 2, \ldots, r$, sînt implicanți primi și se înscriu în tabelul $\mathbf{T}_p$. Implicanții neînscriși în $\mathbf{T}_p$ după această operație se înscriu în tabelul $\mathbf{T}$, iar vectorii $\mathbf{p}_i$ corespunzători în tabelul $\mathbf{T}_0$ și se reia procesul de la pasul 3. Dacă toți implicanții $(\mathbf{s}_i, \mathbf{p}_i)$ obținuți în pasul 10 au fost primi, procesul încetează.

Exemplu. Să se simplifice funcția

$$f(a, b, c) = a^1b^0c^0 + a^1b^1c^0 + a^1b^2c^0 + a^1b^0c^1 + a^1b^1c^1 + a^1b^2c^1 + a^1b^0c^2 + a^1b^1c^2 + a^1b^2c^2$$

folosind metoda celui mai mare divizor comun.

Vectorii asociați constituenților unității sînt: $\mathbf{P}_1 = (1, 0, 0)$,

$\mathbf{P}_2 = (1, 1, 0)$, $\mathbf{P}_3 = (1, 2, 0)$, $\mathbf{P}_4 = (1, 0, 1)$, $\mathbf{P}_5 = (1, 1, 1)$, $\mathbf{P}_6 = (1, 2, 1)$.

$\mathbf{P}_7 = (1, 0, 2)$, $\mathbf{P}_8 = (1, 1, 2)$, $\mathbf{P}_9 = (1, 2, 2)$.

2. Implicanții corespunzători sînt $(\boldsymbol{\theta}, \mathbf{P}_1)$, $(\boldsymbol{\theta}, \mathbf{P}_2)$, $(\boldsymbol{\theta}, \mathbf{P}_3)$, $(\boldsymbol{\theta}, \mathbf{P}_4)$, $(\boldsymbol{\theta}, \mathbf{P}_5)$, $(\boldsymbol{\theta}, \mathbf{P}_6)$, $(\boldsymbol{\theta}, \mathbf{P}_7)$, $(\boldsymbol{\theta}, \mathbf{P}_8)$, $(\boldsymbol{\theta}, \mathbf{P}_9)$ care se înscriu în tabelul $\mathbf{T}$.

$\mathbf{P}^k$ 3. Deoarece toți vectorii $\mathbf{s}_i$ sînt egali cu $\boldsymbol{\theta}$, există o singură submulțime egală cu $\mathbf{P}$.

4. Numerele convenționale asociate constituenților unității și deci vectorilor de mai sus, folosind ponderile 1 pentru a, 3 pentru b și 9 pentru c, sînt:

$$Q_1 = 1,\ Q_2 = 4,\ Q_3 = 7,\ Q_4 = 10,\ Q_5 = 13,\ Q_6 = 16,\ Q_7 = 19,$$
$$Q_8 = 22,\ Q_9 = 25.$$

Înscriem deci vectorii $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, $\mathbf{P}_5$, $\mathbf{P}_6$, $\mathbf{P}_7$, $\mathbf{P}_8$, $\mathbf{P}_9$ în tabelul $\mathbf{T}_0$ în această ordine, care corespunde cu ordinea numerelor convenționale.

5. Se consideră vectorul $\mathbf{P}_9$. Vecinii lui $\mathbf{P}_9$ incluși în $\mathbf{T}_0$ sînt $\mathbf{P}_6$ și $\mathbf{P}_8$. Se înscriu în tabelul $\mathbf{T}_1$ cvasifețele $(\mathbf{P}_6, \mathbf{P}_9)$ și $(\mathbf{P}_8, \mathbf{P}_9)$.

6. Cel mai mare divizor comun al lui P_6 și P_8 este vectorul P_5. Se înscrie în tabelul T_2 cvasifața $(P_6 \cap P_8, P_9) = (P_5, P_9)$ eliminîndu-se din T_1 cvasifețele (P_6, P_9) și (P_8, P_9). Toate cvasifețele din T_1 au fost eliminate și, deoarece T_2 conține o singură cvasifață (P_5, P_9), care este maximală, aceasta se înscrie în T_c. Se reia procesul de la pasul 5 pentru P_8.

5. Vecinii lui P_8 sînt P_5 și P_7. Se înscriu în tabelul T_1 cvasifețele (P_5, P_8) și (P_7, P_8).

6. Cel mai mare divizor comun al lui P_5 și P_7 este vectorul P_1. Se înscrie în tabelul T_2 cvasifața $(P_5 \cap P_7, P_8) = (P_1, P_8)$, eliminîndu-se din T_1 cvasifețele (P_5, P_8) și (P_7, P_8). Cvasifața (P_1, P_8) este maximală și se înscrie în tabelul T_c. Se reia procesul de la pasul 5 pentru P_7.

5. Singurul vecin al lui P_7 este P_4. Se înscrie în tabelul T_1 cvasifața (P_4, P_7). Cvasifața (P_4, P_7) se include în cvasifața (P_4, P_8), deci nu se înscrie în tabelul T_c. Se reia procesul de la pasul 5 pentru P_6.

5. Vecinii lui P_6 sînt P_3 și P_5. Se înscriu în tabelul T_1 cvasifețele (P_3, P_6) și (P_5, P_6).

6. Cel mai mare divizor comun al lui P_3 și P_5 este vectorul P_2. Se înscrie în tabelul T_2 cvasifața $(P_3 \cap P_5, P_6) = (P_2, P_6)$, eliminîndu-se din tabelul T_1 cvasifețele (P_3, P_6) și (P_5, P_6). Cvasifața (P_2, P_6) este maximală și se înscrie în tabelul T_c. Se reia procesul de la pasul 5 pentru P_5.

5. Vecinii lui P_5 sînt P_4 și P_2. Se înscriu în tabelul T_1 cvasifețele (P_2, P_5) și (P_4, P_5).

6. Cel mai mare divizor comun al vectorilor P_2 și P_4 este vectorul P_1. Se înscrie în tabelul T_2 cvasifața $(P_2 \cap P_4, P_5) = (P_1, P_5)$, eliminîndu-se din T_1 cvasifețele (P_2, P_5) și (P_4, P_5), cvasifața (P_1, P_5) este maximală și se înscrie în tabelul T_c. Se reia procesul de la pasul 5 pentru P_4.

5. Singurul divizor al lui P_4 este P_1. Cvasifața (P_1, P_4) este inclusă în cvasifața (P_1, P_5) înscrisă deja în tabelul T_c, astfel încît se trece la pasul următor. Se reia procesul pentru P_3.

6. Vecin al lui P_3 este doar vectorul P_2; cvasifața (P_2, P_3) este inclusă în cvasifața (P_2, P_6), astfel încît nu se înscrie în tabelul T_c. Se reia procesul pentru P_2.

6. Divizorul lui P_2 este P_1, dar, deoarece cvasifața (P_1, P_2) este inclusă în cvasifața (P_1, P_5) aceasta nu se înscrie în tabelul T_c. Întrucît toate elementele din T_0 au fost luate în considerare, se trece la pasul 10.

10. În tabelul T_c au fost incluse cvasifețele (P_5, P_9), (P_4, P_8), (P_2, P_6) și (P_1, P_5) la care corespund implicanții

$$(P_5 \oplus P_9,\ P_5),\ (P_4 \oplus P_8,\ P_4),\ (P_2 \oplus P_6,\ P_2),\ (P_1 \oplus P_5,\ P_1)$$

care se scriu explicit astfel:

$$(s,\ P_5) = ((0,\ 1,\ 1),\ (1,\ 1,\ 1)),\ (s,\ P_4) = ((0,\ 1,\ 1),\ (1,\ 0,\ 1)),$$

$$(s,\ P_2) = ((0,\ 1,\ 1,),\ (1,\ 1,\ 0)),\ (s,\ P_1) = ((0,\ 1,\ 1),\ (1,\ 0,\ 0)).$$

Toți implicanții existenți în T se includ în implicanții (s, P_1), (s, P_2), (s, P_4) și (s, P_5). Îi vom elimina deci din T și în locul lor vor fi înscriși implicanții (s, P_1), (s, P_2), (s, P_4) și (s, P_5).

12. Se reia procesul de la pasul 3.

4^1. Se înscriu în tabelul T_0 numerele convenționale P_1, P_2, P_4 și P_5 corespunzătoare implicanților obținuți în prima etapă a procesului în ordinea (s, P_1), (s, P_2), (s, P_4), (s, P_5).

5^1. Se consideră vectorul P_5. Vecinii lui P_5 sînt P_2 și P_4. Se înscriu în tabelul T_1 cvasifețele (P_2, P_5) și (P_4, P_5).

6. Cel mai mare divizor comun al lui P_2 și P_4 este P_1. Se înscrie în tabelul T_2 cvasifața $(P_2 \cap P_4, P_5) = (P_1, P_5)$, eliminîndu-se din T_1 cvasifețele (P_2, P_5), (P_4, P_5). Cvasifața (P_1, P_5) fiind maximală, se înscrie în T_c.

Implicantul corespunzător cvasifeței (P_1, P_5) este

$$(s + (P_1 \oplus P_5),\ P_1) = (\,(0,\ 2,\ 2),\ (1,\ 0,\ 0)\,),$$

adică

$$a^1(b^0 + b^1 + b^2)(c^0 + c^1 + c^2) = a^1.$$

TEORIA ABSTRACTĂ A AUTOMATELOR

8.1. Generalităţi

Înainte de a trece la problema sintezei automatelor finite, care constituie obiectivul principal al acestei cărţi, vom da cîteva noţiuni de teoria abstractă a automatelor. Cu această ocazie vom preciza clasa de automate de care ne ocupăm şi modul în care înţelegem abordarea problemelor legate de aceste automate. O caracteristică a cărţii este faptul că metodele de sinteză şi de analiză utilizate se aplică automatelor generale, cunoscute în literatura de specialitate sub denumirea de automate Mealy. Automatele Moore vor fi considerate numai ca exemplificări, deşi ar putea să se facă o teorie separată a acestor automate avînd în vedere gradul lor mare de generalitate. Am ales această soluţie ţinînd seama de faptul că un astfel de automat poate fi considerat ca un automat Mealy particular şi poate fi tratat ca atare. Ceea ce este important este că pentru fiecare automat Mealy se poate construi un automat Moore echivalent cu el, dar în general cu un număr mai mare de stări.

O altă caracteristică a acestei cărţi o constituie faptul că întotdeauna din mulţimea de automate echivalente, care corespund unor condiţii de lucru date, se alege automatul sau automatele minime, dintr-un anume punct de vedere, care, în diferitele etape ale sintezei poate fi numărul minim de stări, numărul minim de contacte etc.

Tot cu scopul de a se da un caracter cît mai general metodei de sinteză utilizate, s-au construit algoritmi aplicabili atît automatelor complet definite cît şi automatelor incomplet definite, adică aplicabili cazurilor generale întîlnite în practică.

8.2. Automate Mealy

Definiţia 1. Se numeşte *automat Mealy* un obiect

$$\mathscr{A} = \mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda),$$

compus din trei mulţimi nevide $\mathbf{Z}$, $\mathbf{X}$, $\mathbf{Y}$ şi două funcţii

$$\delta = \delta(\mathbf{z}, \mathbf{x}), \quad \lambda = \lambda(\mathbf{z}, \mathbf{x})$$

definite pe aceste mulţimi. Mulţimea $\mathbf{Z}$ se numeşte *mulţimea stărilor* automatului, mulţimea $\mathbf{X}$ — *mulţimea semnalelor de intrare*, iar mulţimea $\mathbf{Y}$ — *mulţimea semnalelor de ieşire* ale automatului $\mathscr{A}$. Funcţia $\delta(\mathbf{z}, \mathbf{x})$ poartă numele de *funcţie de tranziţie* a automatului. Ea stabileşte o aplicaţie a mulţimii $\mathbf{Z}\times\mathbf{X}$ în mulţimea $\mathbf{Z}$. O astfel de funcţie se mai numeşte *funcţie reflexă* sau *funcţie memorie*. Funcţia $\lambda = \lambda(\mathbf{z}, \mathbf{x})$ se numeşte *funcţie de ieşiri* a automatului $\mathscr{A}$. Ea stabileşte o aplicaţie a mulţimii $\mathbf{Z} \times \mathbf{X}$ în mulţimea $\mathbf{Y}$. O funcţie de această formă se mai numeşte *funcţie de transcodaj*.

Automatele Mealy se vor numi *automate generale* sau, simplu, *automate*. Automatul $\mathscr{A}$ se numeşte *automat iniţial*, dacă în mulţimea $\mathbf{Z}$ a stărilor lui este fixat un element oarecare z_0, numit starea iniţială a automatului $\mathscr{A}$.

Definiţia 2. Un *automat* se numeşte *finit*, dacă sînt finite toate cele trei mulţimi ale sale $\mathbf{Z}$, $\mathbf{X}$, $\mathbf{Y}$.

Pot fi definite şi alte tipuri de finitudine ale automatelor, cum ar fi de exemplu automatele $(\mathbf{X}, \mathbf{Y})$-*finite*, în care sînt finite mulţimea semnalelor de intrare şi mulţimea semnalelor de ieşire, iar mulţimea stărilor poate fi arbitrară. Deşi în această lucrare ne interesează numai automatele finite, menţionăm [5] că noţiunea de automat introdusă

prin definiția 1 este mai generală decît cea introdusă de Mealy prin faptul că mulțimile **Z**, **X**, **Y** pot fi luate arbitrare, și nu în mod obligatoriu mulțimi finite.

Fie $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ și $\mathbf{Y} = \{y_1, y_2, \ldots, y_m\}$ mulțimea semnalelor de intrare și respectiv a semnalelor de ieșire ale automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$. Convenim să numim elementele mulțimii **X** litere de intrare, iar elementele mulțimii **Y** litere de ieșire. De asemenea, vom numi mulțimea **X** alfabetul de intrare, iar mulțimea **Y** alfabetul de ieșire. Vom numi elementele mulțimii $\mathbf{Z} = \{z_1, z_2, \ldots, z_l\}$ litere de stare, iar pe **Z** însăși alfabetul stărilor.

Automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ i se vor atașa trei semigrupuri libere $\mathscr{F}(\mathbf{X})$, $\mathscr{F}(\mathbf{Y})$ și $\mathscr{F}(\mathbf{Z})$ generate de mulțimile **X**, **Y** și **Z** ale semnalelor de intrare, semnalelor de ieșire și stărilor sale. Elementele semigrupului liber $\mathscr{F}(\mathbf{X})$, numit semigrup de intrare, vor fi șiruri finite de litere din **X**, denumite cuvinte de intrare. Elementele semigrupului $\mathscr{F}(\mathbf{Z})$, numit semigrupul stărilor, vor fi șiruri finite de stări din **Z**, denumite cuvinte de stare ale automatului.

Cu aceste definiții se va face o extindere naturală a domeniului de definiție al funcției de tranziție $\delta(\mathbf{z}, \mathbf{x})$ și al funcției de ieșire $\lambda(\mathbf{z}, \mathbf{x})$ ale automatului, înlocuind mulțimea $\mathbf{Z} \times \mathbf{X}$ prin mulțimea $\mathbf{Z} \times \mathscr{F}(\mathbf{X})$.

Fie $p = x_1 x_2 \ldots x_k$ un cuvînt de intrare arbitrar cuprins în $\mathscr{F}(\mathbf{X})$ și o stare arbitrară $z \in \mathbf{Z}$. Vom defini $\delta(z, p)$ astfel :

$$\delta(z, p) = z_1 z_2 \ldots z_k,$$

unde $z_1 = \delta(z, x_1)$, $z_2 = \delta(z_1, x_2)$, $\ldots$, $z_k = \delta(z_{k-1}, x_k)$. De asemenea vom defini $\lambda(z, p)$, după cum urmează :

$$\lambda(z, p) = y_1 y_2 \ldots y_k,$$

unde $y_1 = \lambda(z, x_1)$, $y_2 = \lambda(z_1, x_2)$, $\ldots$, $y_k = \lambda(z_{k-1}, x_k)$.

Definiția 3. Automatul $\mathscr{B}(\mathbf{Z}_1, \mathbf{X}_1, \mathbf{Y}_1, \delta_1, \lambda_1)$ se numește *subautomat* al automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$, dacă $\mathbf{Z}_1 \subset \mathbf{Z}$, $\mathbf{X}_1 \subset \mathbf{X}$, $\mathbf{Y}_1 \subset \mathbf{Y}$ iar funcțiile δ_1 și λ_1 coincid respectiv cu funcțiile δ și λ pe submulțimea $\mathbf{Z}_1 \times \mathbf{X}_1$ a lui $\mathbf{Z} \times \mathbf{X}$.

O categorie interesantă de subautomate este constituită de subautomatele pentru care $\mathbf{Z}_1 \subset \mathbf{Z}$, $\mathbf{X}_1 = \mathbf{X}$, $\mathbf{Y}_1 = \mathbf{Y}$.

8.3. Reprezentările automatelor

Vom da cîteva reprezentări ale automatelor utilizate mai des în literatura de specialitate. Prima dintre acestea este reprezentarea electrică, numită de obicei schema electrică. Nu vom insista asupra acestei reprezentări întrucît în lucrare nu ne interesează modurile particulare de realizare a automatelor. Unele exemple de astfel de reprezentări au fost date în cap. 5 unde s-a tratat problema funcțiilor booleene și funcțiilor de variabile polivalente. Se vor mai da astfel de reprezentări atunci cînd se vor studia cazuri concrete de automate.

A doua reprezentare care va fi utilizată în această lucrare, destul de rar dealtfel, este reprezentarea prin grafuri a automatelor. Convenim să asociem un graf unui automat după următoarele reguli : a) mulțimii stărilor automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ i se ascociază nodurile grafului $\mathscr{G}$, b) dacă stările z_1 și z_2 sînt legate prin relația $z_2 = \delta(z_1, x)$, atunci nodul z_1 este legat cu nodul z_2 printr-un arc orientat de la z_1 spre z_2. Arcul (z_1, z_2) astfel obținut va fi marcat cu x. Evident, este posibil ca să existe mai multe semnale de intrare $x_1, x_2, \ldots, x_p$ pentru care $z_2 = \delta(z_1, x_i)$, $i \in \{1, 2, \ldots, p\}$. În loc să ducem p arce de la z_1 la z_2 vom duce un singur arc pe care îl vom marca cu fiecare din literele $x_1, x_2, \ldots, x_p$. Dacă $y = \lambda(z_1, x)$, atunci arcul (z_1, z_2) va fi marcat și cu litera y și, în general, dacă $x_1, x_2, \ldots, x_p$ verifică relația $y_i = \lambda(z_1, x_i)$, $i \in \{1, 2, \ldots, p\}$, atunci arcul (z_1, z_2) va fi marcat cu fiecare din literele $y_1, y_2, \ldots, y_p$.

Exemplul 1. Automatul cu două semnale de intrare x_1, x_2, două semnale de ieșire y_1, y_2 și patru stări z_1, z_2, z_3, z_4, pentru care funcțiile de tranziție și de ieșiri sînt date de relațiile :

$$\delta(z_1, x_1) = z_2, \qquad \delta(z_1, x_2) = z_3,$$
$$\delta(z_2, x_1) = z_1, \qquad \delta(z_2, x_2) = z_3,$$
$$\delta(z_3, x_1) = z_4, \qquad \delta(z_3, x_2) = z_1,$$
$$\delta(z_4, x_1) = z_2, \qquad \delta(z_4, x_2) = z_4,$$

$$\lambda(z_1, x_1) = y_1, \qquad \lambda(z_1, x_2) = y_2,$$
$$\lambda(z_2, x_1) = y_1, \qquad \lambda(z_2, x_2) = y_1,$$
$$\lambda(z_3, x_1) = y_2, \qquad \lambda(z_3, x_2) = y_2,$$
$$\lambda(z_4, x_1) = y_2, \qquad \lambda(z_4, x_2) = y_1$$

se reprezintă, după metoda descrisă, pe graful din fig. 8.1.

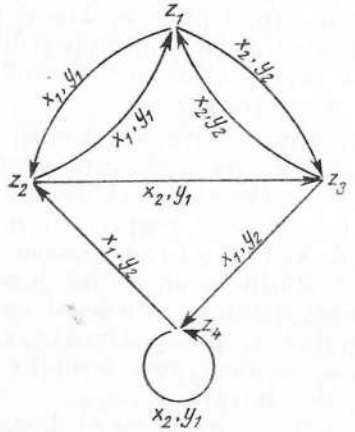

Fig. 8.1

Se observă că automatul din exemplul 1 are o singură stare stabilă z_1 căreia în graful asociat îi corespunde un arc care pleacă din z_1 și se închide în z_1, sub comanda x_2.

A treia și cea mai des utilizată în această lucrare, este metoda tabelară de reprezentare a automatelor. Convenim să asociem unui automat $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ un tabel $\mathbf{T}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ construit astfel :

a) liniile tabelului corespund stărilor automatului ;

b) coloanele tabelului corespund semnalelor de intrare ;

c) dacă z_i este una din stările automatului și x_j unul dintre semnalele de intrare ale acestuia, atunci în tabelul $\mathbf{T}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$, la intersecția liniei i cu coloana j se vor scrie starea $\delta(z_i, x_j)$ și semnalul de ieșire dat de $\lambda(z_i, x_j)$, la fel cum s-a procedat în [1].

Exemplul 2. Automatul dat în exemplul 1 se reprezintă prin metoda tabelară descrisă mai înainte, prin tabelul 8.1.

Tabelul 8.1

Starea \ Intrarea	x_1	x_2
z_1	z_2 y_1	z_3 y_2
z_2	z_1 y_1	z_3 y_1
z_3	z_4 y_2	z_1 y_2
z_4	z_2 y_2	z_4 y_1

Nu vom insista mai mult asupra reprezentării automatelor, întrucît nu este important pentru scopul urmărit în lucrare, însă vom reveni cu unele completări atunci cînd se vor studia cazuri concrete de automate.

8.4. Omomorfismul și echivalența automatelor

8.4.1. *Omomorfismul automatelor*

Presupunem că funcțiile $\delta(z, x)$ și $\lambda(z, x)$ sînt definite pentru mulțimea tuturor perechilor $(z, x) \in \mathbf{Z} \times X$.

Definiția 1. Automatele pentru care funcțiile δ și λ sînt definite pentru orice pereche $(z, x) \in \mathbf{Z} \times \mathbf{X}$ se numesc *automate complet definite* sau, simplu, automate complete. Dimpotrivă, automatele pentru care există cel puțin o pereche $(z, x) \in \mathbf{Z} \times \mathbf{X}$ pentru care funcția δ sau funcția λ nu este definită se numesc *automate incomplet definite* sau, simplu, automate incomplete.

Deocamdată definițiile pe care le vom da se vor referi la automatele complete.

Definiția 2. Un omomorfism ψ al automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ în automatul $\mathscr{B}(\mathbf{Z}_1, \mathbf{X}_1, \mathbf{Y}_1, \delta_1, \lambda_1)$ este dat de mulțimea de trei aplicații ψ_1: $\mathbf{Z} \to \mathbf{Z}_1$, ψ_2 : $\mathbf{X} \to \mathbf{X}_1$ și

$\psi_3 : \mathbf{Y} \to \mathbf{Y}_1$ care satisfac, pentru orice pereche $(z, x) \in \mathbf{Z} \times \mathbf{X}$, relațiile :

$$\psi_1(\delta(z, x)) = \delta_1(\psi_1(z), \psi_2(x)),$$
$$\psi_3(\lambda(z, x)) = \lambda_1(\psi_1(z), \psi_2(x)).$$

Un omomorfism biunivoc a două automate $\mathcal{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ și $\mathcal{B}(\mathbf{Z}_1, \mathbf{X}_1, \mathbf{Y}_1, \delta_1, \lambda_1)$ va fi un izomorfism al acestor automate.

Evident, pot fi întîlnite diferite tipuri de omomorfisme în funcție de cele trei aplicații care intervin în definiția 2. De exemplu, pentru automatele la care coincid mulțimea semnalelor de intrare $\mathbf{X} = \mathbf{X}_1$ și mulțimea semnalelor de ieșire $\mathbf{Y} = \mathbf{Y}_1$, ca aplicații ψ_2 și ψ_3 pot fi luate aplicațiile identice, astfel că relațiile de omomorfism se reduc la următoarele relații :

$$\psi_1(\delta(z, x)) = \delta_1(\psi_1(z), x),$$
$$\lambda(z, x) = \lambda_1(\psi_1(z), x), \quad (z, x) \in \mathbf{Z} \times \mathbf{X}.$$

În acest caz omomorfismul automatelor $\mathcal{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ și $\mathcal{B}(\mathbf{Z}_1, \mathbf{X}_1, \mathbf{Y}_1, \delta_1, \lambda_1)$ se reduce la o singură aplicație $\psi_1 : \mathbf{Z} \to \mathbf{Z}_1$. Un astfel de omomorfism este un omomorfism după stare. La fel se pot defini omomorfisme după intrări, ieșiri sau combinații ale celor trei mulțimi care intervin în definiția automatelor.

Exemplul 3. Să considerăm automatele din tabelele 8.2 și 8.3. Se observă că aplicația ψ care realizează următoarea corespondență a stărilor celor două automate :

$$\psi(z_1) = t_1, \quad \psi(z_4) = t_4, \quad \psi(z_7) = t_7, \quad \psi(z_{10}) = t_8$$
$$\psi(z_2) = t_2, \quad \psi(z_5) = t_5, \quad \psi(z_8) = t_3, \quad \psi(z_{11}) = t_6,$$
$$\psi(z_3) = t_3, \quad \psi(z_6) = t_6, \quad \psi(z_9) = t_7 \quad \psi(z_{12}) = t_1,$$

este un omomorfism după stare al automatului din tabelul 8.2 în automatul din tabelul 8.3, întrucît verifică relațiile de omomorfism. Funcțiile δ_1 și λ_1 ale automatului din tabelul 8.3 sînt date de relațiile :

$$\delta_1(t, x) = \psi(\delta(z, x)), \quad \lambda_1(t, x) = \lambda(z, x),$$

unde $t = \psi(z)$, δ și λ fiind funcțiile corespunzătoare ale automatului din tabelul 8.2.

Tabelul 8.2

	x_1	x_2	x_3	x_4
z_1	z_1, y_1	z_2, y_3	z_6, y_4	z_3, y_2
z_2	z_4, y_1	z_2, y_3	z_5, y_4	z_3, y_2
z_3	z_1, y_1	z_7, y_3	z_6, y_4	z_3, y_2
z_4	z_4, y_1	z_7, y_3	z_5, y_4	z_8, y_2
z_5	z_1, y_1	z_9, y_3	z_5, y_4	z_3, y_2
z_6	z_{12}, y_1	$z_2 y_3$	z_6, y_4	z_{10}, y_3
z_7	z_1, y_1	z_7, y_3	z_{11}, y_4	z_3, y_2
z_8	z_{12}, y_1	z_9, y_3	z_{11}, y_4	z_8, y_2
z_9	z_{12}, y_1	z_9, y_3	z_6, y_4	z_8, y_2
z_{10}	z_1, y_1	z_7, y_3	z_6, y_4	z_{10}, y_3
z_{11}	z_1, y_1	z_2, y_3	z_{11}, y_4	z_{10}, y_3
z_{12}	z_{12}, y_1	z_2, y_3	z_{11}, y_4	z_8, y_2

Tabelul 8.3

	x_1	x_2	x_3	x_4
t_1	$t_1 y_1$	$t_2 y_3$	$t_6 y_4$	$t_3 y_2$
t_2	$t_4 y_1$	$t_2 y_3$	$t_5 y_4$	$t_3 y_2$
t_3	$t_1 y_1$	$t_7 y_3$	$t_6 y_4$	$t_3 y_2$
t_4	$t_4 y_1$	$t_7 y_3$	$t_5 y_4$	$t_3 y_2$
t_5	$t_1 y_1$	$t_7 y_3$	$t_5 y_4$	$t_3 y_2$
t_6	$t_1 y_1$	$t_2 y_3$	$t_6 y_4$	$t_8 y_3$
t_7	$t_1 y_1$	$t_7 y_3$	$t_6 y_4$	$t_3 y_2$
t_8	$t_1 y_1$	$t_7 y_3$	$t_6 y_4$	$t_8 y_3$

8.4.2. *Echivalența automatelor*

Vom introduce noțiunea de echivalență a automatelor complete pe o cale diferită de cea pe care o vom folosi pentru automatele incomplete. Această cale, oarecum naturală, este strîns legată de noțiunea de omomorfism al automatelor și de mulțimea de aplicații ale semigrupului de intrare în semigrupul de ieșire al automatelor, deci este legată de ideea intuitivă care urmărește cunoașterea succesiunii semnalelor de ieșire, generate de o succesiune dată a semnalelor de intrare. Dar tocmai din cauză că este

dificil să se facă o descriere exhaustivă a tuturor elementelor semigrupului de intrare și a elementelor corespunzătoare ale semigrupului de ieșire, această metodă prezintă unele dificultăți.

Mai întîi vom introduce următoarea definiție.

Definiția 1. Se numește *aplicație* φ_z, indusă de starea $z \in \mathbf{Z}$ a automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ o aplicație a semigrupului său de intrare în semigrupul de ieșire, care pune în corespondența unui cuvînt arbitrar $p = x_1 x_2 \dots x_k$ de intrare cuvîntul de ieșire : $\varphi_z(p) = \lambda(z, p)$.

Se numește *familie de aplicații* induse de automatul $\mathscr{A}$ mulțimea tuturor aplicațiilor φ_z diferite două cîte două, $z \in \mathbf{Z}$.

Este natural atunci să introducem următoarea definiție a echivalenței a două stări ale aceluiași automat sau a două automate care au același semigrup de intrare și același semigrup de ieșire.

Definiția 2. Două sau mai multe stări ale unui automat sau ale mai multor automate se numesc *echivalente* dacă induc aceleași *aplicații*.

În fine, mai departe vom putea introduce noțiunea de echivalență a automatelor.

Definiția 3. Două automate $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ și $\mathscr{B}(\mathbf{Z}_1, \mathbf{X}, \mathbf{Y}, \delta_1, \lambda_1)$ care au aceleași alfabete de intrare și de ieșre se numesc *echivalente* dacă induc aceleași *familii de aplicații*.

Următoarea teoremă ne dezvăluie strînsa legătură între noțiunile de omomorfism și de echivalență [5].

Teorema 1. *Dacă există un omomorfism ψ al automatului $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ pe automatul $\mathscr{B}(\mathbf{Z}_1, \mathbf{X}, \mathbf{Y}, \delta_1, \lambda_1)$, atunci automatele $\mathscr{A}$ și $\mathscr{B}$ sînt echivalente. Pentru orice stare $z \in \mathbf{Z}$ stările $z \in \mathbf{Z}$, $\psi(z) \in \mathbf{Z}_1$ sînt echivalente.*

Demonstrație. Să demonstrăm mai întîi echivalența stărilor $z \in \mathbf{Z}$, $\psi(z) \in \mathbf{Z}_1$. Să notăm cu $p = x_1 x_2 \dots x_k$ un cuvînt de intrare oarecare, cu $\delta(z, p) = z_1 z_2 \dots z_k$ cuvîntul de stare care corespunde lui p în starea z a automatului $\mathscr{A}$ și cu $\delta_1(t, p) = t_1 t_2 \dots t_k$, cuvîntul de stare care-i corespunde în starea $t = \psi(z)$ a lui $\mathscr{B}$. Pe baza definiției omomorfismului avem :

$$\psi(z_1) = \psi(\delta(z, x_1)) = \delta_1(\psi(z), x_1) = \delta_1(t, x_1) = t_1,$$
$$\psi(z_2) = \psi(\delta(z_1, x_2)) = \delta_1(\psi(z_1), x_2) = \delta_1(t_1, x_2) = t_2,$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$\psi(z_k) = \psi(\delta(z_{k-1}, x_k)) = \delta_1(\psi(z_{k-1}), x_k) = \delta_1(t_{k-1}, x_k) = t_k.$$

Dacă notăm cu $q = \lambda(z, p) = y_1 y_2 \dots y_k$ cuvîntul de ieșire care corespunde cuvîntului de intrare p în starea z a automatului $\mathscr{A}$, iar cu $q' = \lambda_1(t, p) = y'_1\ y'_2 \dots y'_k$ cuvîntul de ieșire care corespunde cuvîntului de intrare p în starea $t = \psi(z)$ a automatului $\mathscr{B}$, atunci pe baza definiției omomorfismului se obține :

$$y_1 = \lambda(z, x_1) = \lambda_1(\psi(z), x_1) = \lambda_1(t, x_1) = y'_1,$$
$$y_2 = \lambda(z_1, x_2) = \lambda_1(\psi(z_1), x_2) = \lambda_1(t_1, x_2) = y'_2,$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$y_k = \lambda(z_{k-1}, x_k) = \lambda_1(\psi(z_{k-1}), x_k) = \lambda_1(t_{k-1}, x_k) = y'_k,$$

adică

$$q = y_1 y_2 \dots y_k = y'_1\ y'_2 \dots y'_k = q',$$

ceea ce ținînd seama de faptul că p este un cuvînt arbitrar de intrare conduce la concluzia că stările z și $t = \psi(z)$ sînt echivalente. Pe de altă parte, ținînd seama că ψ este un omomorfism, rezultă că pentru fiecare stare z a automatului $\mathscr{A}$ există o stare $t = \psi(z)$ a automatului $\mathscr{B}$ care induce aceeași aplicație $\varphi_t = \varphi_z$. Ținînd seama de faptul că ψ este nelacunar, rezultă că și reciproc, pentru orice stare t automatului $\mathscr{B}$ există o stare z, cu $\psi(z) = t$, care induce aceeași aplicație $\varphi_z = \varphi_t$. Dar atunci automatele $\mathscr{A}$ și $\mathscr{B}$ induc aceeași familie de aplicații și conform definiției 3 sînt echivalente.

Exemplu. Se verifică ușor că automatele din tabelele 8.2 și 8.3 sînt echivalente deoarece

$$\lambda(z, x) = \lambda_1(\psi(z), x) = \lambda_1(t, x)$$

pentru orice $x \in \mathbf{X}$, $z \in \mathbf{Z}$, de unde rezultă că orice cuvînt de intrare $p = u_1 u_2 \dots u_k$ va genera în cele două automate același cuvînt de ieșire :

$$q = v_1 v_2 \dots v_k \lambda = (s_1, u_1)\ \lambda(s_2, u_2) \dots \lambda(s_k, u_k) =$$
$$= \lambda_1(\psi(s_1), u_1)\ \lambda_1(\psi(s_2), u_2) \dots \lambda_1(\psi(s_k), u_k),$$

$s = s_1 s_2 s_3 \ldots s_k$, unde $s_2 = \delta(s_1, u_1)$, $s_3 = \delta(s_2, u_2), \ldots, s_k = \delta(s_{k-1}, u_{k-1})$, s_1 fiind dat odată cu $p = u_1 u_2 \ldots u_k$. În particular, pentru

$$p = x_1 x_1 x_2 x_3 x_4 x_1, \quad s_1 = z_6$$

se obțin cuvintele de ieșire și de stare următoare :

$$q = y_1 y_1 y_3 y_4 y_2 y_1, \quad s = z_6 z_{12} z_{12} z_2 z_5 z_3.$$

Aplicînd omomorfismul ψ, se obține

$$t = \psi(s) = \psi(z_6)\,\psi(z_{12})\,\psi(z_{12})\,\psi(z_2)\,\psi(z_5)\,\psi(z_3)$$

sau $t = t_6 t_1 t_1 t_2 t_5 t_3$, la care corespunde cuvîntul de ieșire

$$\lambda(t, p) = \lambda(t_6, x_1)\,\lambda(t_1, x_1)\,\lambda(t_1, x_2)\,\lambda(t_2, x_3)\,\lambda(t_5, x_4)\,\lambda(t_3, x_1) = y_1 y_1 y_3 y_4 y_2 y_1,$$

adică tot q.

8.5. Automate Moore

Un caz particular interesant de automate Mealy îl constituie automatele Moore.

Definiția 1. Un automat $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ va fi numit *automat Moore* dacă funcția de ieșire $\lambda(z, x)$ poate fi reprezentată sub forma

$$\lambda(z, x) = \mu(\delta(z, x)), \qquad (z, x) \in \mathbf{Z} \times \mathbf{X},$$

unde $\delta(z, x)$ este funcția de tranziție a automatului, iar $\mu(z)$ este o funcție care aplică mulțimea $\mathbf{Z}$ a stărilor automatului $\mathscr{A}$ în mulțimea semnalelor de ieșire. Funcția $\mu(z)$ se numește *funcția de marcaj* iar valoarea $\mu(z)$ a funcției în starea z se numește *marcajul acestei stări*.

Deși constituie cazuri particulare de automate Mealy, automatele Moore au caracter destul de general și se bucură de proprietatea cu totul remarcabilă dată de teorema următoare [5].

Teoremă. *Pentru orice automat Mealy $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta\ \lambda)$ există un automat Moore $\mathscr{B}$, echivalent cu $\mathscr{A}$, iar în cazul cînd $\mathscr{A}$ este finit, automatul $\mathscr{B}$ poate fi ales și el finit cu numărul de stări egal cu $(m + 1)\,n$, unde m este numărul semnalelor de intrare, iar n este numărul stărilor în automatul $\mathscr{A}$.*

Demonstrație. Fiecărei stări z a automatului $\mathscr{A}$ îi punem în corespondență mulțimea stărilor $(w, z_\alpha = (a, x_\alpha))$, unde x_α parcurge mulțimea semnalelor de intrare $\mathbf{X}$. Toate stările obținute în acest fel le luăm ca stări ale automatului $\mathscr{B}$. Funcția de tranziție δ_1 a automatului $\mathscr{B}$ o definim în felul următor :

$$\delta_1 = (w_1, x_\beta) = (z, x_\beta),\ \delta_1((z, x_\alpha), x_\beta) = (\delta(z, x_\alpha), x_\beta).$$

Să arătăm că automatul astfel construit este un automat Moore. Într-adevăr, dacă punem în corespondență stării $b = (z, x_\beta)$ a automatului $\mathscr{B}$ semnalul de ieșire $\lambda(z, x_\beta)$ și dacă notăm cu $\mu(b)$ funcția care realizează această corespondență, atunci

$$\lambda_1(w, x_\beta) = \lambda(z, x_\beta) = \mu(\delta_1(w, x_\beta)),$$

$$\lambda_1((z, x_\alpha), x_\beta) = \lambda(\delta(z, x_\alpha), x_\beta) = \mu(\delta_1(z, x_\alpha), x_\beta).$$

Prin urmare, automatul astfel definit este un automat Moore.

Vom arăta în continuare că automatul $\mathscr{B}$ este echivalent cu automatul $\mathscr{A}$, pe baza căreia s-a construit automatul $\mathscr{B}$.

Fie $z_1 z_2 \ldots z_k$ cuvîntul stărilor iar $y_1 y_2 \ldots y_k$ cuvîntul de ieșire care se pun în corespondență prin starea z a automatului $\mathscr{A}$ unui cuvînt de intrare arbitrar $p = x_1 x_2 \ldots x_k$. Din definiția funcțiilor δ_1 și λ_1 rezultă că starea w a automatului $\mathscr{B}$ pune în corespondență cuvîntului de intrare p, cuvîntul stărilor $b = b_1 b_2 \ldots b_k$, definit astfel :

$$b_1 = \delta_1(w_1, x_1) = (z, x_1),$$

$$b_2 = \delta_1(b_1, x_2) = \delta_1((z, x_1), x_2) = (\delta(z, x_1), x_2) = (z_1, x_2),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$b_k = \delta_1(b_{k-1}, x_k) = \delta_1((z_{k-2}, x_{k-1}), x_k) =$$

$$= (\delta(z_{k-2}, x_{k-1}), x_k) = (z_{k-1}, x_k)$$

și cuvîntul de ieșire $y'_1 y'_2 \dots y'_k$ definit de :

$$y'_1 = \lambda_1(w, x_1) = \lambda(z, x_1) = y_1,$$
$$y'_2 = \lambda_1(b_1, x_2) = \lambda_1((z, x_1), x_2) = \lambda(\delta(z, x_1), x_2) =$$
$$= \lambda(z_1, x_2) = y_2,$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$y'_k = \lambda_1(b_{k-1}, x_k) = \lambda_1((z_{k-2}, x_{k-1}), x_k) =$$
$$= \lambda(\delta(z_{k-2}, x_{k-1}), x_k) = \lambda(z_{k-1}, x_k) = y_k,$$

de unde rezultă că cele două cuvinte de ieșire $y_1 y_2 \dots y_k$ al automatului $\mathscr{A}$ și $y_1 y_2 \dots y_k$ al automatului $\mathscr{B}$ coincid, ceea ce datorită arbitrariului alegerii cuvîntului de intrare $p = x_1 x_2 \dots x_k$ conduce la echivalența stărilor w și z. Apoi, deoarece unul și același semnal de intrare transformă stări echivalente în stări echivalente, rezultă că pentru orice $x_\alpha \in \mathbf{X}$, starea (z, x_α) a automatului $\mathscr{B}$ este echivalentă cu starea $\delta(z, x_\alpha)$ a automatului $\mathscr{A}$. În felul acesta pentru orice stare a automatului $\mathscr{B}$ există o stare a automatului $\mathscr{A}$ echivalentă cu ea. Cum este adevărată și reciproca acestei afirmații, rezultă că automatele $\mathscr{A}$ și $\mathscr{B}$ sînt echivalente, ceea ce demonstrează teorema.

Pentru automatele Moore se pot introduce noțiuni speciale de omomorfism și echivalență Moore. Întrucît scopul acestei lucrări este să tratăm automatele generale și de fapt numai sinteza automatelor generale, nu vom trata această problemă, mulțumindu-ne să o semnalăm. Arătăm de asemenea că prin omomorfism automatele Moore se pot transforma în automate generale, proprietatea unui automat de a fi automat Moore nepăstrîndu-se, în general, prin omomorfism. Vom exemplifica această afirmație în capitolul în care vom trata problema reducerii numărului de stări al automatelor.

În continuare vom da un exemplu de automate Moore omomorfe.

Exemplu. Automatele din tabelele 8.2 și 8.3 sînt automate de tip Moore. Într-adevăr, se verifică imediat că pentru automatul din tabelul 8.2 există funcția $\mu(z)$ definită de relațiile :

$$\mu(z_1) = y_1,\ \mu(z_2) = y_3,\ \mu(z_3) = y_2,\ \mu(z_4) = y_1,\ \mu(z_5) = y_4,$$
$$\mu(z_6) = y_4,\ \mu(z_7) = y_3,\ \mu(z_{10}) = y_3,\ \mu(z_{11}) = y_4,\ \mu(z_{12}) = y_1,$$
$$\mu(z_8) = y_2,\ \mu(z_9) = y_3.$$

Pentru automatul din tabelul 8.3 există funcția $\mu(t)$ definită de relațiile :

$$\mu(t_1) = y_1,\ \mu(t_2) = y_3,\ \mu(t_3) = y_2,\ \mu(t_4) = y_1,\ \mu(t_5) = y_1,\ \mu(t_6) = y_4,$$
$$\mu(t_7) = y_3, \qquad \mu(t_8) = y_3.$$

Rezultă că cele două automate sînt automate Moore.

8.6. Automate incomplete

În 8.2, cînd am definit automatele generale, am presupus că funcțiile $\delta = \delta(z, x)$, $\lambda = \lambda(z, x)$ sînt definite pentru orice pereche $(z, x) \in \mathbf{Z} \times \mathbf{X}$ și iau valori în mulțimea stărilor $\mathbf{Z}$ și respectiv în mulțimea ieșirilor $\mathbf{Y}$. Așa cum am arătat în 8.4, un astfel de automat se numește automat complet definit sau automat complet.

Prin definiție automatele pentru care există cel puțin o pereche $(z, x) \in \mathbf{Z} \times \mathbf{X}$ pentru care funcția $\delta = \delta(z, x)$ sau funcția $\lambda = \lambda(z, x)$ nu este definită se numește automat incomplet definit sau simplu automat incomplet.

Pentru studiul automatelor incomplete nu vom mai putea folosi noțiunile de omomorfism și de echivalență introduse în studiul automatelor complete. În locul noțiunii de echivalență vom folosi în cazul automatelor incomplete noțiunea de compatibilitate, care este mai slabă decît prima, din cauza lipsei axiomei tranzitivității.

O consecință imediată va fi că împărțirea în clase de compatibilitate a mulțimii stărilor automatului nu va conduce la partiții ale mulțimii $\mathbf{Z}$, ci la submulțimi oarecare, fiind posibil ca două astfel de submulțimi să conțină aceeași stare.

După cum se va vedea acest lucru nu va constitui un inconvenient în simplificarea automatelor și nici în stabilirea echivalenței automatelor incomplete.

Să considerăm un automat arbitrar cu n stări dat de tabelul 8.4, unde pentru prescurtarea notațiilor am notat $\delta_{ij} = \delta(z_i, x_j)$ și $\lambda_{ij} = \lambda(z_i, x_j)$ pentru orice $i \in \{1, 2, \dots, n\}$ și orice $j \in \{1, 2, \dots, r\}$.

Tabelul 8.4

	x^1	...	x^j	...	x^r
z_1	$\delta_{11}\ \lambda_{11}$	...	$\delta_{1j}\ \lambda_{1j}$	...	$\delta_{1r}\ \lambda_{1r}$
⋮					
z_i	$\delta_{i1}\ \lambda_{i1}$	...	$\delta_{ij}\ \lambda_{ij}$	...	$\delta_{ir}\ \lambda_{ir}$
⋮					
z_k	$\delta_{k1}\ \lambda_{k1}$	...	$\delta_{kj}\ \lambda_{kj}$	...	$\delta_{kr}\ \lambda_{kr}$
⋮					
z_n	$\delta_{n1}\ \lambda_{n1}$	...	$\delta_{nj}\ \lambda_{nj}$	...	$\delta_{rn}\ \lambda_{rn}$

Fie $\mathbf{Z}_{ij} \subset \mathbf{Z} \times \mathbf{Z}$ mulțimea de perechi de stări din $\mathbf{Z}$ definită astfel:

$$\mathbf{Z}_{ij} = \{(\delta_{ik}, \delta_{jk}),\ k \in \mathbf{K}_{ij},\ \text{unde}\ \mathbf{K}_{ij} \subset \{1, 2, \ldots, r\},$$

este mulțimea de indici pentru care sînt definite atît starea δ_{ik} cît și starea δ_{jk}, i și j fiind dați.

Definiția 1. Submulțimea $\mathbf{Z}_{ij}$ va fi denumită *submulțimea de perechi de stări asociate* perechii (z_i, z_j) sau, simplu, mulțimea asociată perechii (z_i, z_j).

Exemplul 1. Pentru automatul cu matricea de tranziție a stărilor și de ieșire din tabelul 10.2 se obțin submulțimile asociate: $\mathbf{Z}_{12} = \{(z_3, z_5)\}$, $\mathbf{Z}_{13} = O$, $\mathbf{Z}_{14} = \{(z_3, z_3)\}$ etc.

Definiția 2. Două stări z_i și z_j sînt *compatibile* dacă submulțimea asociată $\mathbf{Z}_{ij}$ îndeplinește una oricare dintre condițiile: a) este mulțimea vidă; b) conține perechea (z_i, z_j) și/sau perechi (z_k, z_k), $k \in \{1, 2, \ldots, n\}$, n fiind numărul de stări; c) conține perechi de stări compatibile, iar ieșirile îndeplinesc una oricare dintre condițiile: a) $\lambda_{ik} = \lambda_{jk}$, $k \in \{1, 2, \ldots, r\}$; b) una oricare dintre ieșiri este nedefinită; c) amîndouă ieșirile sînt nedefinite.

Două stări z_i și z_j pentru care mulțimile $\mathbf{Z}_{ij}$ asociate îndeplinesc punctele a) și b) referitoare la perechile de stări și punctele a), b) și c) referitoare la ieșiri se vor numi *stări direct compatibile.*

De exemplu, în automatul reprezentat în tabelul 10.2 stările z_1 cu z_3 și z_1 cu z_4 sînt direct compatibile.

Definiția 3. Două stări z_i și z_j sînt *incompatibile* dacă verifică una oarecare dintre condițiile: a) există un indice $k \in \{1, 2, \ldots, r\}$ astfel încît $\lambda_{ik} \neq \lambda_{jk}$, λ_{ik} și λ_{jk} amîndouă definite, b) există două stări z_r și z_s incompatibile, $(z_r, z_s) \in \mathbf{Z}_{ij}$. Stările care îndeplinesc condiția a) se vor numi *direct incompatibile.*

De exemplu, în automatul din tabelul 10.2 starea z_4 este direct incompatibilă cu stările z_2 și z_3, la fel cum starea z_6 este direct incompatibilă cu stările z_1, z_2, z_4 și z_5.

Definiția 4. Două stări z_i și z_j sînt *nedeterminate* dacă ieșirile îndeplinesc condițiile de compatibilitate dar există cel puțin o pereche $(z_r, z_s) \in \mathbf{Z}_{ij}$, $(z_r, z_j) \neq (z_i, z_j)$, $r \neq s$.

Stările z_i și z_j nedeterminate pot fi compatibile sau incompatibile, dar nu se poate decide acest lucru apriori. De exemplu, stările z_1 și z_2 ale automatului din tabelul 10.2 sînt nedeterminate, întrucît $\mathbf{Z}_{12}$ conține perechea $(z_3, z_5) \neq (z_1, z_2)$.

Lemă. *Relația de compatibilitate este reflexivă și simetrică.*

Demonstrație. Reflexivitatea rezultă din faptul că mulțimea $\mathbf{Z}_{ii}$ asociată perechii de stări (z_i, z_i), $i \in \{1, 2, \ldots, n\}$, conține numai perechi de forma (z_j, z_j), $j \in \{1, 2, \ldots, n\}$, iar ieșirile λ_{ij} sînt fie nedefinite fie sînt aceleași, deci ne găsim în cazul b) din definiția 2.

Simetria rezultă din faptul că $\mathbf{Z}_{ij} = \mathbf{Z}_{ji}$ pentru orice $i, j \in \{1, 2, \ldots, n\}$.

Vom da o metodă de recunoaștere a stărilor compatibile atunci cînd vom trata problema simplificării automatelor. Tot atunci vom demonstra că relația de compatibilitate astfel introdusă, pentru automatele incomplete, devine relația de echivalență pentru automatele complete. Această proprietate este foarte importantă și ne permite să extindem metodele de studiu ale automatelor incomplete la automate complete și să considerăm automatele com-

plete drept cazuri particulare de automate incomplete. Metodele pe care le vom utiliza în studiul automatelor incomplete capătă astfel un caracter general.

Exemplu. Ca exemplu de automat incomplet vom da discriminatorul de sens de rotație înscris în tabelul 8.5, în care pentru fiecare stare există un semnal de intrare pentru care nu este definit nici δ, nici λ.

Tabelul 8.5

	x_1	x_2	x_3	x_4
z_1	z_1 1	z_3 1		z_8 0
z_2	z_2 0	z_3 1		z_8 0
z_3	z_2 0	z_3 1	z_5 1	
z_4	z_2 0	z_4 0	z_5 1	
z_5		z_4 0	z_5 1	z_7 1
z_6		z_4 0	z_6 1	z_7 1
z_7	z_1 1		z_6 0	z_7 1
z_8	z_1 1		z_6 0	z_8 0

Tabelul 8.6

	x_1	x_2	x_3	x_4
t_1	t_1 1	t_2 1	t_4 0	t_1 0
t_2	t_2 0	t_2 1	t_3 1	t_1 0
t_3	t_2 0	t_3 0	t_3 1	t_4 1
t_4	t_1 1	t_3 0	t_4 0	t_4 1

Se va demonstra mai departe că automatul din tabelul 8.5 este echivalent cu automatul complet din tabelul 8.6, care se obține din tabelul 8.5 prin fuzionarea stărilor compatibile:

$$t_1 = (z_1, z_8),\ t_2 = (z_2, z_3),\quad t_3 = (z_4, z_5),\ t_4 = (z_6, z_7).$$

În plus trebuie remarcat că automatul din tabelul 8.6 este un automat Mealy complet, în timp ce automatul din tabelul 8.5 este un automat Moore care se poate scrie ca în tabelul 8.7, unde în ultima coloană am trecut valorile funcției de ieșire $\lambda(z_i, x_j)$. Se vede că se poate scrie $\lambda(z_i, x_j) = \varphi(z_i)$, unde $\varphi(z_i) = 1$ pentru i impar și $\varphi(z_i) = 0$ pentru i par.

Ori de cîte ori automatul din tabelul 8.7 ajunge în starea z_i, acesta emite semnalul $\varphi(z_i)$ pe cînd automatul din tabelul 8.6 emite semnale care depind atît de starea în care se ajunge, cît și de semnalul de intrare care a determinat tranziția.

Tabelul 8.7

z \ x	x_1	x_2	x_3	x_4	y
z_1	z_1	z_3		z_8	1
z_2	z_2	z_3		z_8	0
z_3	z_2	z_3	z_5		1
z_4	z_2	z_4	z_5		0
z_5		z_4	z_5	z_7	1
z_6		z_4	z_6	z_7	0
z_7	z_1		z_6	z_7	1
z_8	z_1		z_6	z_8	0

CAPITOLUL 9

PRINCIPIUL METODEI LUI HUFFMAN

9.1. Punerea problemei

În acest capitol se vor prezenta principiile metodei lui Huffman de sinteză a automatelor finite [35].

După Huffman sinteza automatelor finite cuprinde următoarele șase etape:

1° asignarea stărilor stabile;
2° stabilirea matricei de faze primitive;
3° stabilirea matricei de faze reduse;
4° afectarea variabilelor intermediare și stabilirea matricei de excitații secundare;
5° condiția tehnică de bună funcționare, eliminarea hazardului din circuitele de comutație;
6° stabilirea matricei de ieșire.

Pentru fiecare dintre aceste etape există diferite metode de rezolvare.

9.2. Asignarea stărilor stabile

Asignarea stărilor stabile ale unui automat se poate face prin mai multe metode, în funcție de felul cum este dată problema. De obicei, în această etapă sînt preponderente metodele intuitive de asignare. Prima fază din

această etapă constă, de regulă, în enumerarea stărilor stabile și transpunerea rezultatului analizei problemei într-o diagramă de tranziție sau într-o matrice de faze. În metoda lui Huffman se folosește ultima dintre acestea, rezultatul analizei transpunîndu-se în matricea de faze primitivă.

Vom reveni, mai departe, asupra metodelor de asignare a stărilor. În acest capitol vom expune metoda lui Huffman pe care o vom ilustra printr-un exemplu.

Presupunem că dorim să construim un automat în care variabilele de intrare x_1, x_2 și semnalul de ieșire y au funcționarea descrisă de diagrama de timp din fig. 9.1.

Fiecare dintre variabilele x_1, x_2 și y are cîte două stări. Notînd cu 1 starea de funcționare, reprezentată prin linia continuă și prin 0 starea de nefuncționare, rezultă că există patru combinații ale variabilelor de intrare (00, 01, 11, 10) care reprezintă tot atîtea semnale de intrare.

Definiție. Prin semnale de intrare (ieșire) înțelegem *stările variabilelor de intrare (ieșire)*. O succesiune de semnale de intrare (ieșire) este o succesiune de stări ale variabilelor de intrare (ieșire). O variabilă de intrare (ieșire) este o variabilă asociată unui element fizic care poate fi contactul unui buton, un circuit poartă, o lampă de semnalizare etc.

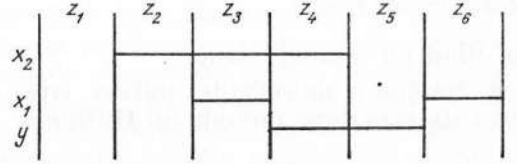

Fig. 9.1

În schema din fig. 9.1 există patru semnale de intrare (00, 01, 11, 10), valorile perechii x_1x_2 și două semnale de ieșire (0, 1), valorile variabilei y. Întrucît în diagrama de timp din fig. 9.1 intervin numai semnalele de intrare și de ieșire ale automatului căutat, stările acestui automat pot fi exprimate în funcție de acestea. Ele corespund fa-

zelor diferite, notate cu z_1, z_2, z_3, z_4, z_5 și z_6 în fig. 9.1. Fiecărei stări îi corespunde o pereche (x_1x_2, y), astfel: $z_1 = (00, 0)$, $z_2 = (10, 0)$, $z_3 = (11, 0)$, $z_4 = (10, 1)$, $z_5 = (00, 1)$, $z_6 = (01, 0)$. Se obține și restul de elemente necesare pentru construcția funcției de tranziție $\delta(z, x_1x_2)$:

$$\delta(z_1, 00) = z_1, \quad \delta(z_1, 01) = z_2;$$

$$\delta(z_2, 00) = z_1, \quad \delta(z_2, 01) = z_2, \quad \delta(z_2, 11) = z_3;$$

$$\delta(z_3\ 01) = z_2, \quad \delta(z_3, 11) = z_3, \quad \delta(z_3, 10) = z_4;$$

$$\delta(z_4, 11) = z_3, \quad \delta(z_4, 10) = z_4;$$

$$\delta(z_5, 00) = z_5, \quad \delta(z_5, 10) = z_6;$$

$$\delta(z_6, 00) = z_1, \quad \delta(z_6, 10) = z_6$$

și funcției de ieșiri, $\lambda(z, x_1x_2)$:

$$\lambda(z_1, x_1x_2) = \mu(z_1) = 0,$$

$$\lambda(z_2, x_1x_2) = \mu(z_2) = 0,$$

$$\lambda(z_3, x_1x_2) = \mu(z_3) = 0,$$

$$\lambda(z_4, x_1x_2) = \mu(z_4) = 1,$$

$$\lambda(z_5, x_1x_2) = \mu(z_5) = 1,$$

$$\lambda(z_6, x_1x_2) = \mu(z_6) = 0$$

automatul fiind un automat Moore.

Pe baza relațiilor precedente putem trece la faza următoare a sintezei după metoda lui Huffman.

9.3. Stabilirea matricei de faze primitive

Pentru simplificarea scrierii și pentru a ne încadra în notațiile utilizate în descrierea metodei lui Huffman, în continuare vom nota stările prin numere, convenind să notăm starea z_k prin numărul k. Folosind metoda tabelară de reprezentare a automatelor Moore din relațiile

anterioare, vom obține tabelul 9.1, în care am încercuit stările stabile și am notat cu y funcția $\mu(z_k)$. Acest tabel

Tabelul 9.1

z \ x_1x_2	00	01	11	10	y
1	①	2			0
2	1	②	3		0
3		2	③	4	0
4	5		3	④	1
5	⑤			6	1
6				⑥	0

reprezintă o matrice în care liniile sînt fazele automatului dat și care în metoda lui Huffman a fost denumită matricea de fază primitivă. Ea prezintă o serie de avantaje în studiul simplificării automatelor și în asignarea elementelor intermediare și finale, ceea ce justifică larga ei utilizare în studiul automatelor.

Stabilirea matricei de faze primitive obligă la o analiză completă a problemei, în timp ce metodele intuitive pot să lase anumite aspecte interesante ale problemei în afara analizei.

9.4. Stabilirea matricei de faze reduse

Matricea de faze primitivă este rezultatul unei analize care urmărește o descriere completă a automatului, bazată de regulă pe descompunerea în faze elementare. În acest proces de enumerare se pot introduce, sub numere diferite, stări care în fapt nu sînt diferite sau care sînt compatibile. Reducerea matricelor de faze primitive se face prin eliminarea stărilor redondante și prin fuzionarea stărilor compatibile. Există diferite metode pentru rezolvarea acestei etape.

Dacă există două stări caracterizate prin faptul că la aceleași intrări corespund aceleași ieșiri și că orice tranziție duce la stări stabile identice sau echivalente, atunci una din strările respective este redondantă și poate fi eliminată. Se convine să fie eliminată linia cu număr mai mare. Se observă ușor că în tabelul 9.1 perechile

de stări (1, 2), (1, 3), (1, 6), (2, 3), (2, 6) se bucură de proprietatea precedentă. Mai mult, mulțimea de stări (1, 2, 3) în ansamblul ei se bucură de această proprietate.

Dacă notăm cu $\mathscr{R}$ relația indusă de această proprietate, atunci evident $\mathscr{R}$ este o relație de echivalență. Deci stările 1, 2, 3 pot fi înlocuite cu una singură, fie starea 1, și automatul din tabelul 9.1 se reduce la automatul din tabelul 9.2.

Tabelul 9.2

z \ x_1x_2	00	01	11	10	y
1	①	①	①	4	0
4	5		1	④	1
5	⑤			6	1
6				⑥	0

Reducerea matricelor de faze primitive prin eliminarea stărilor redondante este o fază care transformă automate Moore în automate Moore. Am arătat în cap. 8 că în general transformările de echivalență care păstrează automatele Moore nu conduc la automate minime. De aceea trebuie continuat procesul de reducere prin eliminarea stărilor compatibile. Procesul poate fi realizat într-un singur pas, deoarece, evident, stările redondante sînt în același timp compatibile.

Exemplu. Să se construiască matricea de faze redusă pentru automatul din fig. 9.2, unde nodurile reprezintă stările, arcele tranzițiile, variabilele notate pe arce comenzile, iar numerele înscrise în cercuri alături de stări sînt ieșirile.

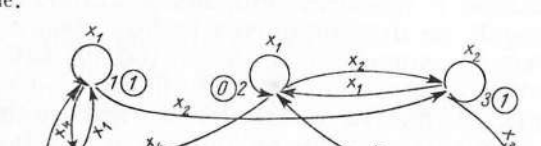

Fig. 9.2.

Matricea de faze primitivă va fi cea din tabelul 9.3. Am marcat ca deobicei stările stabile prin încercuire. Se observă că tabelul 9.3 reprezintă automatul din tabelul 8.7 cu stările notate altfel și deci este echivalent cu automatul din tabelul 8.6 obținut prin fuzionarea perechilor de stări compatibile : 1 cu 8, 2 cu 3, 4 cu 5 și 6 cu 7. Păstrînd convenția de înlăturare, în cazul stărilor compatibile, a stărilor cărora le corespund numere mai mari, se obține automatul din tabelul 9.4, echivalent cu automatul 9.3.

Tabelul 9.3

z \ x	x_1	x_2	x_3	x_4	y
1	①	3		8	1
2	②	3		8	0
3	2	③	5		1
4	2	④	5		0
5		4	⑤	7	1
6		4	⑥	7	0
7	1		6	⑦	1
8	1		6	⑧	0

Tabelul 9.4

z \ xy	x_1	y	x_2	y	x_3	y	x_4	y
1	①	1	2	1	6	0	①	0
2	②	0	②	1	4	1	1	0
4	2	0	④	0	④	1	6	1
6	1	1	4	0	⑥	0	⑥	1

9.5. Afectarea variabilelor intermediare și stabilirea matricei de excitații secundare

În ipoteza utilizării de elemente bipoziționale, ipoteză pe care o vom folosi, un automat cu N stări poate fi realizat cu un număr minim de M elemente, unde M este dat de relația

$$2^{M-1} < N \leqslant 2^M.$$

Din punct de vedere teoretic există un număr foarte mare de posibilități de realizare, chiar pentru automate cu număr redus de stări. Este clar că, din acest punct de vedere, se poate asocia fiecărei stări a automatului orice combinație a variabilelor intermediare, cu condiția ca pentru stări diferite să se asocieze combinații diferite. Prin combinații ale variabilelor intermediare înțelegem stări ale acestor variabile. Nu folosim expresia stare a variabilelor intermediare pentru a nu confunda aceste stări cu stările automatului.

Numărul de combinații diferite ale variabilelor intermediare este $A^N_{2^M}$, unde $A^N_{2^M}$ este numărul de aranjamente a 2^M elemente luate cîte N, M este numărul de variabile intermediare iar N numărul de stări ale automatului. De exemplu, automatul din tabelul 9.4 pentru care $N=4$, $M=2$ se poate realiza cu două elemente intermediare. Numărul de realizări posibile ale automatului respectiv este de $A^4_{2^2} = P_4 = 4! = 24$.

Fie **S** mulțimea de combinații posibile ale variabilelor intermediare. O realizare a automatului $\mathcal{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ în **S** este o aplicație bijectivă a mulțimii **Z** a stăriilor automatului pe o parte $\mathbf{S}' \in \mathbf{S}$. Fie γ această aplicație: $\mathbf{Z} \xrightarrow{\gamma} \mathbf{S}'$. Prin afectarea variabilelor intermediare înțelegem operația de stabilire a unei aplicații γ. Întrucît există un număr mare de posibilități de stabilire a aplicațiilor γ, în alegerea acestor aplicații se vor folosi criterii economice, unele condiții tehnice de realizare etc.

Să reținem din matricea de faze redusă numai partea care privește stările îndepărtînd ieșirile. În aceste condiții, de exemplu, matricea din tabelul 9.4 se reduce la matricea din tabelul 9.5. O astfel de matrice va fi numită *matrice*

Tabelul 9.5

z \ x	x_1	x_2	x_3	x_4
1	①	2	6	①
2	②	②	4	1
4	2	④	④	6
6	1	4	⑥	⑥

de tranziție a stărilor. În particular fiind vorba de o matrice minimală, aceasta va fi numită *matrice redusă de tranziție a stărilor*.

Dacă γ este o aplicație bijectivă a mulțimii stărilor automatului pe $\mathbf{S}'$, prin înlocuirea stărilor automatului cu stările corespunzătoare ale variabilelor intermediare, din matricea redusă de tranziție se obține o matrice pe care o vom numi *matrice de excitații secundare*. Această matrice ne va da funcțiile de lucru ale elementelor intermediare.

Automatul cu matricea de tranziție a stărilor din tabelul 9.5 se poate realiza, după cum am arătat, cu două elemente intermediare bivalente. Fie z_1 și z_2 aceste elemente și fie γ aplicația

$$1 \leftrightarrow 00; \qquad 4 \leftrightarrow 11;$$

$$2 \leftrightarrow 01; \qquad 6 \leftrightarrow 10.$$

Înlocuind în tabelul 9.5, se obține matricea de excitații secundare din tabelul 9.6.

Tabelul 9.6

z_1z_2 \ x	x_1	x_2	x_3	x_4
00	00	01	10	00
01	01	01	11	00
11	01	11	11	10
10	00	11	10	10

Notînd cu $f_\gamma(z)$ funcția de lucru a elementului intermediar z, din tabelul precedent se obține următoarele funcții de lucru pentru z_1 și z_2:

$$f_\gamma(z_1) = \bar{z}_1 z_2 x_1 + z_1 z_2 (x_2 + x_3) + z_1 \bar{z}_2 (x_3 + x_4) + \bar{z}_1 \bar{z}_2 x_1,$$

$$f_\gamma(z_2) = \bar{z}_1 z_2 (x_1 + x_2) + z_1 z_2 (x_2 + x_3) + \bar{z}_1 \bar{z}_2 x_4 + z_1 \bar{z}_2 x_4,$$

care după simplificare se reduc la formele :

$$f_Y(z_1) = \bar{z}_1 x_1 + z_1 z_2 (x_2 + x_3) + z_1 \bar{z}_2 (x_3 + x_4),$$
$$f_Y(z_2) = z_1 z_2 x_1 + z_2 x_2 + \bar{z}_1 z_2 x_3 + \bar{z}_2 x_4.$$

9.6. Condiția tehnică de bună funcționare. Eliminarea hazardului din circuitele de comutație

Am presupus că intrările x_1, x_2, x_3, x_4 sînt la rîndul lor stări ale semnalelor de intrare. Fie $x_1 = 00$, $x_2 = 01$, $x_3 = 11$, $x_4 = 10$. Atunci automatul din tabelul 9.6 se scrie sub forma din tabelul 9.7.

Tabelul 9.7

z_1z_2 \ u_1u_2	00	01	11	10
00	00	01	10	00
01	01	01	11	00
11	01	11	11	10
10	00	11	10	10

Cu u_1, u_2 am notat cele două semnale de intrare. Observînd că se poate scrie $x_1 = \bar{u}_1\bar{u}_2$, $x_2 = \bar{u}_1 u_2$, $x_3 = u_1 u_2$, $x_4 = u_1\bar{u}_2$, funcțiile de lucru $f_Y(z_1)$ și $f_Y(z_2)$ devin :

$$f_Y(z_1) = \bar{z}_1\bar{u}_1\bar{u}_2 + z_1 z_2 (\bar{u}_1 u_2 + u_1 u_2) + z_1\bar{z}_2 (u_1 u_2 + u_2\bar{u}_2) = \\ = \bar{z}_1\bar{u}_1\bar{u}_2 + z_1 z_2 u_2 + z_1\bar{z}_2 u_1,$$

$$f_Y(z_2) = \bar{z}_1 z_2\bar{u}_1\bar{u}_2 + z_2\bar{u}_1 u_2 + \bar{z}_1 z_2 u_1 u_2 + \bar{z}_2 u_1\bar{u}_2 = \bar{z}_1 z_2\bar{u}_1 + \\ + z_2\bar{u}_1 u_2 + \bar{z}_1 z_2 u_2 + \bar{z}_2 u_1\bar{u}_2.$$

Să analizăm funcționarea automatului din tabelul 9.7. Se observă că trecerea de la o stare la alta se face în două faze la variația semnalelor de intrare. Mai întîi la variația unui din semnalele de intrare se face trecerea pe orizontală din starea stabilă inițială într-o stare instabilă, apoi păstrînd nemodificată starea semnalelor de intrare se realizează o funcționare autonomă de trecere a automatului din starea instabilă în starea stabilă corespunzătoare. Se spune că avem o cascadă a stărilor secundare. Se poate verifica ușor că trecerea autonomă dintr-o stare în alta

se face, în toate cazurile, prin variația unei singure variabile intermediare. Se spune că un astfel de automat îndeplinește condiția tehnică de bună funcționare.

Îndeplinirea condiției tehnice de bună funcționare se poate realiza fie prin schimbarea afectării variabilelor intermediare, fie prin introducerea de stări suplimentare. În capitolul următor vom da o metodă de afectare a stărilor care să îndeplinească condiția tehnică de bună funcționare. Aici ne mulțumim să exemplificăm cele două situații.

Să presupunem că afectarea variabilelor intermediare s-ar fi făcut astfel : $1 \leftrightarrow 00$, $2 \rightarrow 11$, $3 \leftrightarrow 01$, $4 \leftrightarrow 10$. Noua afectare diferă de afectarea precedentă numai prin inversarea stărilor 2 și 3. Se va obține automatul din tabelul 9.8. Se observă că dacă elementele z_1 și z_2 nu

Tabelul 9.8

z_1z_2 \ u_1u_2	00	01	11	10
00	00	11	10	00
11	11	11	01	00
01	11	01	01	10
10	00	01	10	10

acționează în același timp, pot interveni erori de funcționare. De exemplu, dacă din starea 00 sub comanda 00 se aplică comanda 01 și dacă elementul z_2 acționează înaintea elementului z_1, în loc să se treacă în starea stabilă 11 se trece în starea 01 și se stabilizează în această stare.

Se spune că în funcționarea automatului apare hazardul. Eliminarea acestui tip de hazard se poate face, după cum am văzut, prin permutarea stărilor 2 și 3, respectiv prin schimbarea afectării variabilelor intermediare.

Următorul exemplu arată că schimbarea afectării variabilelor intermediare nu este satisfăcătoare în toate cazurile pentru eliminarea hazardului. Să considerăm automatul din tabelul 9.9. În acest caz există $A^3_{2^2} = 24$ posibilități de afectare a variabilelor intermediare însă nici o astfel de afectare nu va elimina hazardul, deoarece nici una nu va putea îndeplini condiția tehnică de bună funcționare. Eliminarea hazardului se va putea face prin introducerea unor stări suplimentare, respectiv prin dedublarea unor stări sau afectarea multiplă a elementelor intermediare.

z_1z_2 \ x_1x_2	00	01	11	10
00	00	01	11	00
01	11	01	01	01
11	11	11	11	00

9.7. Stabilirea matricei de ieșire

În paragrafele precedente am stabilit pornind de la matricea de excitații secundare funcțiile de lucru ale variabilelor intermediare. Aici ne propunem să construim funcțiile de lucru ale variabilelor de ieșire plecînd de la matricea de ieșire.

Numim *matrice de ieșire* matricea obținută din matricea de faze redusă prin înlăturarea din aceasta a stărilor. De exemplu, pentru automatul din tabelul 9.4 se obține matricea de ieșire din tabelul 9.10.

z \ x	x_1	x_2	x_3	x_4
1	1	1	0	0
2	0	1	1	0
4	0	0	1	1
6	1	0	0	1

Dacă înlocuim în tabelul 9.10 stările cu stările variabilelor intermediare afectate iar intrările cu stările elementelor de intrare corespunzătoare, se obține, de exemplu,

z_1z_2 \ u_1u_2	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	1	0	0	1

pentru automatul din tabelul 9.7 matricea de ieșire din tabelul 9.11 care conduce la funcția de lucru a elementului de ieșire y :

$$f_Y(y) = \bar{z}_1\bar{z}_2(\bar{u}_1\bar{u}_2 + \bar{u}_1u_2) + \bar{z}_1z_2(\bar{u}_1u_2 + u_1u_2) + z_1z_2(u_1u_2 + u_1\bar{u}_2) + z_1\bar{z}_2(\bar{u}_1\bar{u}_2 + u_1\bar{u}_2)$$

sau

$$f_Y(y) = \bar{z}_1\bar{z}_2\bar{u}_1 + \bar{z}_1z_2u_2 + z_1z_2u_1 + z_1\bar{z}_2\bar{u}_2.$$

MINIMIZAREA NUMĂRULUI DE STĂRI ALE AUTOMATELOR FINITE

10.1. Introducere

În procesul de sinteză a automatelor finite prezintă importanță economică și tehnică problema reducerii numărului de stări. În acest capitol vom prezenta o metodă sistematică de realizare a acestei etape a procesului de sinteză [32]. În principiu, această metodă constă în introducerea unei relații de compatibilitate în mulțimea stărilor automatului, determinarea matricei de compatibilitate a automatului și realizarea, pe această bază, a clasificării stărilor.

Se presupune că avem un automat $\mathcal{A}$ $(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$. în general incomplet definit, unde $\mathbf{Z}$ este mulțimea stărilor, $\mathbf{X}$ mulțimea intrărilor, $\mathbf{Y}$ mulțimea ieșirilor, $\delta : \mathbf{Z} \times \mathbf{X} \to \mathbf{Z}$ funcția de tranziție a stărilor și $\lambda : \mathbf{Z} \times \mathbf{X} \to \mathbf{Y}$ funcția de ieșire (tabelul 10.1).

Tabelul 10.1

z \ x	x_1	...	x_j	...	x_l
z_1	$\delta(z_1, x_1)$, $\lambda(z_1, x_1)$	...	$\delta(z_1, x_j)$ $\lambda(z_1, x_j)$	...	$\delta(z_1, x_l)$ $\lambda(z_1, x_l)$
z_i	$\delta(z_i, x_1)$ $\lambda(z_i, x_1)$	...	$\delta(z_i, x_j)$ $\lambda(z_i, x_j)$	...	$\delta(z_i, x_l)$ $\lambda(z_i, x_l)$
z_k	$\delta(z_k, x_1)$ $\lambda(z_k, x_1)$	...	$\delta(z_k, x_j)$ $\lambda(z_k, x_j)$	...	$\delta(z_k, x_l)$ $\lambda(z_k, x_l)$
z_n	$\delta(z_n, x_1)$ $\lambda(z_n, x_1)$	...	$\delta(z_n, x_j)$ $\lambda(z_n, x_j)$	...	$\delta(z_n, x_l)$ $\lambda(z_n, x_l)$

Conform convenției utilizate în capitolul precedent vom folosi următoarele noțiunii simplificatoare: $\delta_{ij} = \delta(z_i, x_j)$ și $\lambda_{ij} = \lambda(z_i, x_i)$, $z_i \in \mathbf{Z}$, $x_i \in \mathbf{X}$, $\delta_{ij} \in \mathbf{Z}$, $\lambda_{ij} \in \mathbf{Y}$.

10.2. Relația de compatibilitate a stărilor

În 8.6 am introdus noțiunea de compatibilitate a stărilor și de mulțime asociată unei perechi de stări.

Aici vom aprofunda aceste noțiuni și vom studia relația de compatibilitate a stărilor automatului. Pe baza rezultatelor ce se vor obține se va da un algoritm de determinare a matricei de compatibilitate care va fi folosită mai departe la deducerea algoritmului de minimizare a numărului de stări.

Fie $\mathbf{Z}_{ij}$, $i, j \in \{1, 2, \ldots, n\}$, submulțimile de perechi de stări asociate perechilor de stări $(z_i, z_j) \in \mathbf{Z} \times \mathbf{Z}$. Convenim să asociem perechii de stări (z_i, z_j) variabila booleană e_{ij} cu proprietatea $e_{ij} = 1 \Leftrightarrow z_i$, compatibil

Tabelul 10.2

z \ x	x_1	y	x_2	y
z_1			z_3	0
z_2	z_3	0	z_5	0
z_3	z_4	0		
z_4	z_5	1	z_4	0
z_5			z_6	0
z_6				1

cu z_j și $e_{ij} = 0 \Leftrightarrow z_i$, incompatibil cu z_j. Submulțimilor $\mathbf{Z}_{ij}$ asociate perechilor $(z_i, z_j) \in \mathbf{Z} \times \mathbf{Z}$ le asociem expresii booleene $\mathscr{E}(\mathbf{Z}_{ij})$, după cum urmează:

1. $\mathscr{E}(\mathbf{Z}_{ij}) = e_{ij}$, dacă z_i este direct compatibil cu z_j.
2. $\mathscr{E}(\mathbf{Z}_{ij}) = \bar{e}_{ij}$, dacă z_i este direct incompatibil cu z_j.
3. $\mathscr{E}(\mathbf{Z}_{ij}) = e_{ij} \prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} + \bar{e}_{ij} \sum\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} \bar{e}_{rs}$, dacă z_i și z_j sînt nedeterminate.

Vom nota prin $\mathfrak{M}$ mulțimea perechilor de stări direct compatibile, prin $\mathfrak{N}$ mulțimea perechilor de stări direct incompatibile și prin $\mathfrak{P}$ mulțimea perechilor de stări nedeterminate.

Lema 1. *Condiția necesară și suficientă pentru ca două stări z_i și z_j nedeterminate să fie compatibile este dată de egalitatea* $\prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} = 1$. *De asemenea, pentru ca stările z_i și z_j nedeterminate să fie incompatibile este necesar și suficient ca* $\prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} = 0$.

Demonstrație. Într-adevăr, se poate verifica ușor că există următoarele implicații:

a) $\prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} \Rightarrow e_{ij}$;

b) $\overline{\prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs} \Rightarrow \bar{e}_{ij}$.

Într-adevăr, dacă stările z_r și z_s sînt compatibile pentru orice pereche $(z_r, z_s) \in \mathbf{Z}_{ij}$, atunci stările z_i și z_j sînt compatibile conform cu definiția 2, de unde rezultă că se verifică implicația a). Dacă din contră există cel puțin o pereche de stări $(z_r, z_s) \in \mathbf{Z}_{ij}$, incompatibile, atunci stările z_i și z_j sînt incompatibile și deci se verifică implicația b). Cu aceasta am demonstrat suficiența condițiilor din lemă. Să arătăm că ele sînt și necesare.

Într-adevăr, implicațiile a) și b) sînt echivalente cu relația

$$\bar{e}_{ij} \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} + e_{ij} \overline{\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs} = 0$$

și respectiv cu relația

$$e_{ij} \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} + \bar{e}_{ij} \overline{\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs} = 1,$$

adică $e_{ij} = \prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs}$ și $\bar{e}_{ij} = \overline{\prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs}$ și deci $e_{ij} = 1 \Leftrightarrow$

$\Leftrightarrow \prod\limits_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} = 1$ și $\bar{e}_{ij} = 1$, de unde

$$\overline{\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs} = 1 \Leftrightarrow \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} = 0.$$

Să considerăm acum expresia

$$\mathcal{E}(\mathbf{Z}_{ij}) = e_{ij} \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} + \bar{e}_{ij} \overline{\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs}.$$

Lema 2. *Condiția necesară și suficientă pentru ca două stări z_i și z_j să fie compatibile este dată de egalitatea $\mathcal{E}(\mathbf{Z}_{ij}) = e_{ij}$. De asemenea egaliteatea $\mathcal{E}(\mathbf{Z}_{ij}) = \bar{e}_{ij}$ dă condiția necesară și suficientă pentru ca stările z_i și z_j să fie incompatibile.*

Demonstrație. $\mathcal{E}(\mathbf{Z}_{ij}) = e_{ij} \Rightarrow$

$$\Rightarrow \bar{e}_{ij} \overline{\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs} + e_{ij} \overline{\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}}} e_{rs} = 0 \Rightarrow \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} = 1.$$

Reciproc, $\prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} = 1 \Rightarrow \mathcal{E}(\mathbf{Z}_{ij}) = e_{ij}$, ceea ce demonstrează prima parte a lemei.

Partea a doua a lemei se demonstrează similar, astfel că nu vom mai face demonstrația, lăsînd-o pe seama cititorului.

Fie z_i și z_j două stări nedeterminate și $\mathcal{E}(\mathbf{Z}_{ij})$ expresia booleană asociată, $\mathcal{E}(\mathbf{Z}_{ij}) = e_{ij} \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} + \bar{e}_{ij} \sum_{(z_r, z_s) \in \mathbf{Z}_{ij}} \bar{e}_{rs}$.

Vom nota prin $\mathcal{E}^1(\mathbf{Z}_{ij})$ expresia obținută din $\mathcal{E}(\mathbf{Z}_{ij})$ prin egalarea cu 1 a tuturor variabilelor e_{pq} pentru care perechile de stări (z_p, z_q) sînt incluse în $\mathbf{Z}_{ij} \cap \mathfrak{M}$.

Lema 3. *Între expresia booleană $\mathcal{E}(Z_{ij})$ asociată mulțimii de perechi de stări $\mathbf{Z}_{ij}$ și expresia $\mathcal{E}^1(Z_{ij})$ obținută din $\mathcal{E}(\mathbf{Z}_{ij})$ prin egalarea cu 1 a tuturor variabilelor e_{pq} corespunzătoare perechilor de stări $(z_p, z_q) \in \mathbf{Z}_{ij} \cap \mathfrak{M}$ există relația*

$$\prod_{(z_p, z_q) \in \mathbf{Z}_{ij}} e_{pq} \cdot \mathcal{E}(\mathbf{Z}_{ij}) = \prod_{(z_p, z_q) \in \mathbf{Z}_{ij}} e_{pq} \cdot \mathcal{E}^1(\mathbf{Z}_{ij}).$$

Demonstrație. Vom face demonstrația plecînd de la identitatea (a) $xf(x) = xf(1)$, a cărei demonstrație este imediată. Într-adevăr, se poate scrie succesiv : $xf(x) = x[xf(1) + \bar{x}f(0)] = xf(1)$.

Dacă luăm $x = e_{pq}$ și $f(x) = \mathcal{E}(\mathbf{Z}_{ij})$ și dacă ținem seama de notația făcută, $f(1) = \mathcal{E}^1(\mathbf{Z}_{ij})$, atunci se poate scrie

$$e_{pq}\, \mathcal{E}(\mathbf{Z}_{ij}) = e_{pq}\, \mathcal{E}^1(\mathbf{Z}_{ij}).$$

Schimbînd rolul lui $\mathcal{E}(\mathbf{Z}_{ij})$ cu $\mathcal{E}^1(\mathbf{Z}_{ij})$, se poate aplica identitatea (a) folosind o altă variabilă corespunzătoare unei alte perechi de stări din $\mathbf{Z}_{ij} \cap \mathfrak{M}$. Continuînd procesul pentru toate perechile de stări din $\mathbf{Z}_{ij} \cap \mathfrak{M}$, se obține relația din enunțul lemei.

Fie acum $\mathcal{E}^0(\mathbf{Z}_{ij})$ expresia obținută din $\mathcal{E}(\mathbf{Z}_{ij})$ prin egalarea cu 0 a uneia dintre variabilele e_{pq} corespunzătoare perechilor de stări $(z_p, z_q) \in \mathbf{Z}_{ij} \cap \mathfrak{N}$.

Lema 4. *Dacă variabila booleană e_{pq} corespunde unei perechi de stări $(z_p, z_q) \in \mathbf{Z}_{ij} \cap \mathfrak{N}$, atunci există relația*

$$\bar{e}_{pq}\, \mathcal{E}(\mathbf{Z}_{ij}) = \bar{e}_{pq}\, \bar{e}_{ij}.$$

Demonstrație. Vom pleca de la demonstrația identității (b) $\bar{x}f(x) = \bar{x}f(0)$. Se obține succesiv : $\bar{x}f(x) = \bar{x}[xf(1) + \bar{x}f(0)] = \bar{x}f(0)$. Luînd $x = e_{pq}$ și $f(x) = \mathcal{E}(\mathbf{Z}_{ij})$,

Tabelul 10.3

z \ x	x_1	y	x_2	y
1			3	0
2	3	0	5	0
3	4	0		
4	5	1	3	0
5			6	0
6				1

$(z_p, z_q) \in \mathbf{Z}_{ij} \cap \mathfrak{N}$, și aplicînd identitatea (b), se obține imediat $\bar{e}_{pq}\, \mathcal{E}(\mathbf{Z}_{ij}) = \bar{e}_{pq}\, \mathcal{E}^0(\mathbf{Z}_{ij})$. Dar $\mathcal{E}^0(\mathbf{Z}_{ij}) = \bar{e}_{ij}$, de unde rezultă relația din enunțul lemei.

Vom face următoarele notații :

$$\mathcal{E}_{\mathfrak{W}}(\mathbf{Z}) = \prod_{(z_i, z_j) \in \mathfrak{W}} \mathcal{E}(\mathbf{Z}_{ij}), \qquad \mathcal{E}_{\mathfrak{N}}(\mathbf{Z}) = \prod_{(z_i, z_j) \in \mathfrak{N}} \mathcal{E}(\mathbf{Z}_{ij}),$$

$$\mathcal{E}_{\mathfrak{P}}(\mathbf{Z}) = \prod_{(z_i, z_j) \in \mathfrak{P}} \mathcal{E}(\mathbf{Z}_{ij}) \text{ și } \mathcal{E}(\mathbf{Z}) = \mathcal{E}_{\mathfrak{W}}(\mathbf{Z})\, \mathcal{E}_{\mathfrak{N}}(\mathbf{Z})\, \mathcal{E}_{\mathfrak{P}}(\mathbf{Z}).$$

Definiția 4. Expresiile booleene $\mathcal{E}_{\mathfrak{W}}(\mathbf{Z})$, $\mathcal{E}_{\mathfrak{N}}(\mathbf{Z})$ și $\mathcal{E}_{\mathfrak{P}}(\mathbf{Z})$ vor fi denumite *expresiile booleene asociate* respectiv *mulțimii perechilor de stări compatibile, mulțimii perechilor de stări incompatibile* și *mulțimii perechilor de stări nedeterminate.*

Expresia $\mathcal{E}(\mathbf{Z})$ va fi denumită expresia booleană asociată mulțimii stărilor automatului sau, simplu, expresia asociată automatului și va fi notată prin $\mathcal{E}(\mathbf{A})$ atunci cînd nu se ivesc confuzii.

Lema 5. *Expresia $\mathcal{E}(\mathbf{Z})$ asociată mulțimii $\mathbf{Z}$ este separabilă în trei expresii,* $\mathcal{E}(\mathbf{Z}) = \mathcal{E}_1(\mathbf{Z})\,\mathcal{E}_2(\mathbf{Z})\,\mathcal{E}_3(\mathbf{Z})$, *unde* $\mathcal{E}_1(\mathbf{Z}) = \mathcal{E}_{\mathfrak{W}}(\mathbf{Z})$, $\mathcal{E}_2(\mathbf{Z}) = \mathcal{E}_{\mathfrak{N}}(\mathbf{Z})$ *iar* $\mathcal{E}_3(\mathbf{Z})$ *conține numai variabile corespunzătoare perechilor de stări nedeterminate.*

Demonstrație. Conform definiției, $\mathcal{E}_{\mathfrak{W}}(\mathbf{Z})$ și $\mathcal{E}_{\mathfrak{N}}(\mathbf{Z})$ conțin numai variabile asociate perechilor de stări din $\mathfrak{M}$ și respectiv $\mathfrak{N}$. Rămîne să arătăm că $\mathcal{E}_{\mathfrak{P}}(\mathbf{Z})$ poate fi separată conform cerințelor lemei. Conform definiției $\mathcal{E}_{\mathfrak{P}}(\mathbf{Z}) = \prod_{(z_i,\, z_j)\in\mathfrak{P}} \mathcal{E}(\mathbf{Z}_{ij})$. Apoi folosind lema 3, pentru orice pereche $(z_p, z_q)\in\mathfrak{M}$ și orice expresie $\mathcal{E}(\mathbf{Z}_{ij})$, se poate scrie $\prod_{(z_p,\, z_q)\in\mathfrak{W}} e_{pq}\mathcal{E}(\mathbf{Z}_{ij}) = \prod_{(z_p,\, z_q)\in\mathfrak{W}} e_{pq}\cdot\mathcal{E}^1(\mathbf{Z}_{ij})$, unde $\mathcal{E}^1(\mathbf{Z}_{ij}) = e_{ij}$, dacă stările z_i și z_j sînt compatibile, sau $\mathcal{E}^1(\mathbf{Z}_{ij}) = e_{ij}\prod_{(z_r,\, z_s)\in\mathbf{Z}'_{ij}} e_{rs} + \bar{e}_{ij}\sum_{(z_r,\, z_s)\in\mathbf{Z}'_{ij}} \bar{e}_{rs}$, $\mathbf{Z}'_{ij} = \mathbf{Z}_{ij}\cap\mathfrak{M}$, adică $\mathbf{Z}'_{ij}$ nu conține perechi de stări din $\mathfrak{M}$.

La fel, folosind lema 4, pentru orice pereche $(z_p, z_q)\in\mathfrak{N}$ și orice expresie $\mathcal{E}(\mathbf{Z}_{ij})$, se poate scrie $\bar{e}_{pq}\mathcal{E}(\mathbf{Z}_{ij}) = \bar{e}_{pq}\,\mathcal{E}^0(\mathbf{Z}_{ij})$, unde $\mathcal{E}^0(\mathbf{Z}_{ij}) = \bar{e}_{ij}$, dacă $(z_p, z_q)\in\mathbf{Z}_{ij}$ și $\mathcal{E}^0(\mathbf{Z}_{ij}) = \mathcal{E}(\mathbf{Z}_{ij})$ în caz contrar. $\mathcal{E}^0(\mathbf{Z}_{ij})$ conține numai variabile asociate perechilor $(z_i, z_j)\in\mathfrak{P}$.

Mulțimea $\mathfrak{P}$ se poate împărți în trei submulțimi $\mathfrak{P}_1$, $\mathfrak{P}_2$, $\mathfrak{P}_3$ astfel: $\mathfrak{P}_1$ conține perechile de stări din $\mathfrak{P}$, compatibile, $\mathfrak{P}_2$ conține perechile de stări incompatibile, iar $\mathfrak{P}_3$ perechile de stări care rămîn nedeterminate după simplificările de mai înainte. Folosind rezultatele precedente, se poate scrie

$$\mathcal{E}(\mathbf{Z}) = \mathcal{E}_{\mathfrak{W}}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{N}}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{P}}(\mathbf{Z}) = \mathcal{E}_{\mathfrak{W}}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{N}}(\mathbf{Z})\mathcal{E}_{\mathfrak{P}_1}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{P}_2}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{P}_3}(\mathbf{Z}),$$

unde

$$\mathcal{E}_{\mathfrak{P}_1}(\mathbf{Z}) = \prod_{(z_i,\, z_j)\in\mathfrak{P}_1} e_{ij},\quad \mathcal{E}_{\mathfrak{P}_2}(\mathbf{Z}) = \prod_{(z_i,\, z_j)\in\mathfrak{P}_2} \bar{e}_{ij} \text{ și } \mathcal{E}_{\mathfrak{P}_3}(\mathbf{Z}) =$$

$$= \prod_{(z_i,\, z_j)\in\mathfrak{P}_3} \left(e_{ij}\prod_{(z_r,\, z_s)\in\mathbf{Z}'_{ij}} e_{rs} + \bar{e}_{ij}\prod_{(z_r,\, z_s)\in\mathbf{Z}'_{ij}} \bar{e}_{rs}\right),\ \mathbf{Z}'_{ij} = \mathbf{Z}_{ij}\cap\mathfrak{P}_3.$$

Definiția 5. Un *automat* este *determinat* dacă are stările determinate, altfel spus, dacă stările automatului sînt două cîte două compatibile sau incompatibile. Reciproc, un *automat* este *nedeterminat* dacă are stări nedeterminate.

Un automat determinat poate să fie incomplet definit, după cum se poate verifica pe automatul din tabelul 10.4 în care nu este definită trecerea din starea D sub comanda x^2.

Tabelul 10.4

z \ x	x^1		x^2	
K	D	0	C	0
C	C	1	D	0
D	C	0		1

Vom demonstra acum o teoremă care ne conduce la un algoritm pentru minimizarea numărului de stări în automatele finite.

Teoremă. *Condiția necesară și suficientă pentru ca un automat să fie determinat este dată de relația* $\mathcal{E}(\mathbf{Z}) = 1$.

Demonstrație. Din lema 6 rezultă că $\mathcal{E}(\mathbf{Z})$ se poate scrie sub forma:

$$\mathcal{E}(\mathbf{Z}) = \mathcal{E}_{\mathfrak{W}}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{N}}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{P}_1}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{P}_2}(\mathbf{Z})\,\mathcal{E}_{\mathfrak{P}_3}(\mathbf{Z}),\ \mathcal{E}(\mathbf{Z}) = 1 \Leftrightarrow$$

$$\Leftrightarrow (\mathcal{E}_{\mathfrak{W}}(\mathbf{Z}) = 1,\quad \mathcal{E}_{\mathfrak{N}}(\mathbf{Z}) = 1,\quad \mathcal{E}_{\mathfrak{P}_1}(\mathbf{Z}) = 1,\quad \mathcal{E}_{\mathfrak{P}_2}(\mathbf{Z}) = 1 \text{ și}$$

$$\mathcal{E}_{\mathfrak{P}_3}(\mathbf{Z}) = 1, \mathcal{E}_{\mathfrak{W}}(\mathbf{Z}) = 1, \mathcal{E}_{\mathfrak{N}}(\mathbf{Z}) = 1, \mathcal{E}_{\mathfrak{P}_1}(\mathbf{Z}) = 1 \text{ și } \mathcal{E}_{\mathfrak{P}_2}(\mathbf{Z}) = 1$$

sînt condiții necesare și suficiente pentru ca perechile de stări din $\mathfrak{M}$, $\mathfrak{N}$, $\mathfrak{P}_1$ și $\mathfrak{P}_2$ să fie compatibile sau incompatibile, după caz.

Pentru demonstrarea teoremei rămîne să ne ocupăm deci numai de perechile de stări nedeterminate, din $\mathfrak{P}_3$. Presupunem că $\mathscr{E}_{\mathfrak{P}_3}(\mathbf{Z}) = 1$. Atunci

$$\mathscr{E}(\mathbf{Z}'_{ij}) = e_{ij} \prod_{(z_r,\, z_s)\in \mathbf{Z}'_{ij}} e_{rs} + \bar{e}_{ij} \sum_{(z_r,\, z_s)\in \mathbf{Z}'_{ij}} \bar{e}_{rs} = 1, \quad \mathbf{Z}'_{ij} = \mathbf{Z}_{ij} \cap \mathfrak{P}_3,$$

pentru orice i și j pentru care $(z_i, z_j) \in \mathfrak{P}_3$. Dar atunci $e_{ij} = \prod_{(z_r,\, z_s)\in \mathbf{Z}'_{ij}} e_{rs}$ și deci $\mathscr{E}(\mathbf{Z}'_{ij})$ ia fie valoarea e_{ij}, fie valoarea $\bar{e}_{ij}$ și conform lemei 2 stările z_i și z_j sînt determinate. Reciproc, dacă toate perechile de stări din $\mathfrak{P}_3$ sînt determinate, atunci $\mathscr{E}(\mathbf{Z}'_{ij}) = 1$ pentru orice i și j pentru care $(z_i, z_j) \in \mathfrak{P}_3$, deci și $\mathscr{E}_{\mathfrak{P}_3}(\mathbf{Z}) = 1$, ceea ce demonstrează teorema.

Rezultă imediat următoarea consecință a teoremei.

Consecință. *Fiecare constituent al unității al funcției* $\mathscr{E}(\mathbf{Z})$ *conduce la un automat determinat, compatibil cu automatul dat. Mai mult, automatele determinate, astfel obținute, sînt singurele automate determinate, compatibile cu automatul dat.*

Lema următoare ne permite să alegem din mulțimea automatelor determinate pe cel care conduce la automatul redus cu număr minim de stări. Fie $\mathscr{C}(\mathbf{Z})$ mulțimea constituenților unității ai funcției $\mathscr{E}(\mathbf{Z})$. Dacă $\mathbf{C} \in \mathscr{C}(\mathbf{Z})$ este un constituent al unității al lui $\mathscr{E}(\mathbf{Z})$, vom nota prin $\mathscr{M}(\mathbf{C})$ mulțimea elementelor necomplementate din $\mathbf{C}$.

Lema 6. *În mulțimea* $\mathscr{C}(\mathbf{Z})$ *a constituenților unității ai funcției* $\mathscr{E}(\mathbf{Z})$ *există un constituent al unității* $\mathbf{C} \in \mathscr{C}(\mathbf{Z})$, *astfel încît pentru oricare alt constituent al unității* $\mathbf{C}' \in \mathscr{C}(\mathbf{Z})$, *există relația* $\mathscr{M}(\mathbf{C}') \subset \mathscr{M}(\mathbf{C})$.

Demonstrație. Expresia $\mathscr{E}(\mathbf{Z})$ poate fi scrisă astfel:

$$\mathscr{E}(\mathbf{Z}) = \prod_{(z_i,\, z_j)\in \mathfrak{B}\cup\mathfrak{P}_1} e_{ij} \prod_{(z_i,\, z_j)\in \mathfrak{R}\cup\mathfrak{P}_2} \bar{e}_{ij} \prod_{(z_i,\, z_j)\in \mathfrak{P}_3} \mathscr{E}(\mathbf{Z}_{ij}),$$

unde

$$\mathscr{E}(\mathbf{Z}_{ij}) = e_{ij} \prod_{(z_r,\, z_s)\in \mathbf{Z}'_{ij}} e_{rs} + \bar{e}_{ij} \sum_{(z_r,\, z_s)\in \mathbf{Z}'_{ij}} \bar{e}_{rs}, \quad \mathbf{Z}'_{ij} = \mathbf{Z}_{ij} \cap \mathfrak{P}_3.$$

Este suficient să luăm constituentul unității care conține termenul

$$\mathbf{C} = \prod_{(z_i,\, z_j)\in \mathfrak{P}_3} e_{ij} \prod_{(z_r,\, z_s)\in \mathbf{Z}'_{ij}} e_{rs}, \quad \mathbf{Z}'_{ij} = \mathbf{Z}_{ij} \cap \mathfrak{P}_3,$$

care satisface condițiile din lema 6, după cum se poate verifica ușor.

Definiția 6. Monomul lui $\mathscr{E}(\mathbf{Z})$ care îndeplinește condițiile lemei 6 se va numi *element generator al automatului redus* și-l vom nota cu $\mathscr{E}_A$.

Fie $\mathscr{E}_A = \prod_{i,\, j=\{1,\, 2,\, \ldots,\, n\}} e_{ij}^{\alpha_{ij}}$ elementul generator al automatului $\mathscr{A}$, unde n este numărul de stări din $\mathbf{Z}$, iar α_{ij} are valoarea 0 sau 1 după cum variabila e_{ij} apare negată în $\mathscr{E}_A$ sau nu. Vom asocia elementului $\mathscr{E}_A$ matricea

$$\mathfrak{A} = (\alpha_{ij}), \quad i,\, j \in \{1,\, 2,\, \ldots,\, n\},$$

unde convenim să luăm $\alpha_{ii} = 1$, $(\forall)\, i \in \{1,\, 2, \ldots,\, n\}$, iar α_{ij}, pentru $i \neq j$, sînt valorile puterilor lui e_{ij} în $\mathscr{E}_A$.

Definiția 7. Matricea $\mathfrak{A}$ asociată elementului generator $\mathscr{E}_A$ se va numi *matricea de compatibilitate a automatului* $\mathbf{A}$.

Matricea $\mathfrak{A}$ definește relația de compatibilitate dintre stările automatului. Într-adevăr $\alpha_{ij}=1$ corespunde apariției lui e_{ij} în $\mathscr{E}_A$, ceea ce marchează faptul că stările z_i și z_j sînt compatibile, iar $\alpha_{ij} = 0$ corespunde apariției lui $\bar{e}_{ij}$ în $\mathscr{E}_A$, deci cazului în care stările z_i și z_j sînt incompatibile. Vom nota prin $\mathscr{R}(\mathbf{A})$ relația de compatibilitate automatului $\mathbf{A}$, $\mathscr{R}(\mathbf{A}) \subset \mathbf{Z} \times \mathbf{Z}$.

Rezultatele obținute pînă în prezent ne permit să dăm următorul algoritm de determinare a matricei de compatibilitate a unui automat finit $\mathbf{A}$.

Descrierea algoritmului. 1. Se determină mulțimea perechilor de stări direct compatibile $\mathfrak{M}$, mulțimea perechilor de stări direct incompatibile $\mathfrak{N}$ și mulțimea perechilor de stări nedeterminate $\mathfrak{P}$.

2. Pentru fiecare pereche de stări $(z_i, z_j) \in \mathfrak{P}$ se determină mulțimea de prechi de stări asociată $\mathbf{Z}_{ij}$.

3. Se asociază expresiile booleene $\mathscr{E}(\mathbf{Z}_{ij})$ mulțimilor asociate $\mathbf{Z}_{ij}$, după cum urmează:

$$\mathscr{E}(\mathbf{Z}_{ij}) = e_{ij}, \quad (z_i, z_j) \in \mathfrak{M},$$

$$\mathscr{E}(\mathbf{Z}_{ij}) = \bar{e}_{ij}, \quad (z_i, z_j) \in \mathfrak{N},$$

$$\mathscr{E}(\mathbf{Z}_{ij}) = e_{ij} \prod_{(z_r, z_s) \in \mathbf{Z}_{ij}} e_{rs} + \bar{e}_{ij} \sum_{(z_r, z_s) \in \mathbf{Z}_{ij}} \bar{e}_{rs}, \quad (z_i, z_j) = \mathfrak{P}.$$

Determinarea mulțimilor asociate $\mathbf{Z}_{ij}$ pentru perechile de stări din $\mathfrak{M}$ și $\mathfrak{N}$ nu a fost necesară, întrucît aceste mulțimi nu intervin în definirea expresiilor booleene asociate.

4. Se scrie expresia booleană $\mathscr{E}(\mathbf{Z})$ asociată mulțimii stărilor automatului $\mathbf{A}$:

$$\mathscr{E}(\mathbf{Z}) = \prod_{(z_i, z_j) \in \mathbf{Z} \times \mathbf{Z}} \mathscr{E}(\mathbf{Z}_{ij}), \quad i < j,$$

5. Se extrage din $\mathscr{E}(\mathbf{Z})$ elementul generator $\mathscr{E}_A$ al automatului $\mathbf{A}$. Fie

$$\mathscr{E}_A = \prod_{i, j \in \mathbf{I}_n} e_{ij}^{\alpha_{ij}}, \quad i < j, \ \mathbf{I}_n = \{1, 2, \ldots, n\}.$$

6. Se asociază elementelui generator $\mathscr{E}_A$ matricea simetrică

$$\mathfrak{A} = (\alpha_{ij}), \quad \alpha_{ji} = \alpha_{ij}, \quad \alpha_{ii} = 1, \quad i, j \in \{1, 2, \ldots, n\},$$

care este matricea de compatibilitate a automatului $\mathbf{A}$.

Exemplu. Să se determine matricea de compatibilitate pentru automatul din tabelul 10.2.

Vom transcrie automatul în tabelul 10.3, înlocuind stările cu numerele lor.

1. Determinăm mulțimile $\mathfrak{M}$, $\mathfrak{N}$ și $\mathfrak{P}$:

$$\mathfrak{M} = \{(1, 3), (1, 4), (3, 5), (3, 6)\},$$

$$\mathfrak{N} = \{(1, 6), (2, 4), (2, 6), (3, 4), (4, 6), (5, 6)\}.$$

$$\mathfrak{P} = \{(1, 2), (1, 5), (2, 3), (2, 5), (4, 5)\}.$$

2. Determinăm mulțimile asociate $\mathbf{Z}_{ij}$ pentru perechile $(z_i, z_j) \in \mathfrak{P}$:

$\mathbf{Z}_{12} = \{(3, 5)\}$, $\mathbf{Z}_{15} = \{(3, 6)\}$, $\mathbf{Z}_{23} = \{(3, 4)\}$, $\mathbf{Z}_{25} = \{(5, 6)\}$, $\mathbf{Z}_{45} = \{(3, 6)\}$.

3. Asociem expresiile booleene mulțimilor asociate $\mathbf{Z}_{ij}$:

$$\mathscr{E}(\mathbf{Z}_{ij}) = e_{ij}, \ (z_i, z_j) \in \mathfrak{M}, \quad \mathscr{E}(\mathbf{Z}_{ij}) = \bar{e}_{ij}, \ (z_i, z_j) \in \mathfrak{N}.$$

$$\mathscr{E}(\mathbf{Z}_{12}) = e_{12} e_{35} + \bar{e}_{12} \bar{e}_{35},$$

$$\mathscr{E}(\mathbf{Z}_{15}) = e_{15} e_{36} + \bar{e}_{15} \bar{e}_{36}, \quad \mathscr{E}(\mathbf{Z}_{25}) = e_{25} e_{56} + \bar{e}_{25} \bar{e}_{56},$$

$$\mathscr{E}(\mathbf{Z}_{23}) = e_{21} e_{34} + \bar{e}_{23} \bar{e}_{34}, \quad \mathscr{E}(\mathbf{Z}_{45}) = e_{45} e_{36} + \bar{e}_{45} \bar{e}_{36}.$$

4. Scriem expresia booleană $\mathscr{E}(\mathbf{Z})$:

$$\mathscr{E}(\mathbf{Z}) = e_{13} e_{14} e_{35} e_{36} \bar{e}_{16} \bar{e}_{24} \bar{e}_{26} \bar{e}_{34} \bar{e}_{46} \bar{e}_{56}$$

$$(e_{12} e_{35} + \bar{e}_{12} \bar{e}_{35}) (e_{15} e_{36} + \bar{e}_{15} \bar{e}_{36}) (e_{23} e_{34} + \bar{e}_{23} \bar{e}_{34})$$

$$(e_{25} e_{56} + \bar{e}_{25} \bar{e}_{56}) (e_{45} e_{36} + \bar{e}_{45} \bar{e}_{36}) =$$

$$= e_{12} e_{13} e_{14} e_{15} \bar{e}_{16} \bar{e}_{23} \bar{e}_{24} \bar{e}_{25} \bar{e}_{26} \bar{e}_{34} e_{35} e_{36} e_{45} \bar{e}_{46} \bar{e}_{56}.$$

5. Elementul generator al automatului este $\mathscr{E}_A = \mathscr{E}(\mathbf{Z})$.

6. Scriem matricea $\mathfrak{A}$:

$$\mathfrak{A} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

Observație. Pe primul rînd se scriu de obicei stările corespunzătoare coloanelor.

10.3. Algoritmul de clasificare a stărilor

Fie $\mathscr{P}(\mathbf{Z})$ mulțimea părților lui $\mathbf{Z}$ și $\mathbf{X} \in \mathscr{P}(\mathbf{Z})$ o submulțime oarecare a lui $\mathbf{Z}$. Vom nota prin $\mathfrak{A}_X$ matricea obținută din $\mathfrak{A}$ ca intersecție a liniilor și coloanelor corespunzătoare stărilor din $\mathbf{X}$.

Definiția 1. Matricea $\mathfrak{A}_X$ se numește *matricea asociată submulțimii* $\mathbf{X} \subset \mathbf{Z}$.

Vom nota prin $\mathscr{P}(\mathfrak{A})$ următoarea mulțime de matrice:

$$\mathscr{P}(\mathfrak{A}) = \{\mathfrak{A}_X \mid \mathbf{X} \in \mathscr{P}(\mathbf{Z})\},$$

pe care o vom numi familia de matrice asociate mulțimii părților lui **Z**.

Vom demonstra o lemă care ne va folosi la stabilirea algoritmului de clasificare a stărilor automatului.

Fie $\mathbf{X} = \{z_1, z_2, \ldots, z_p\}$ o submulțime de p stări din **Z**.

Lema 1. *Condiția necesară și suficientă pentru ca stările* $z_1, z_2, \ldots, z_p \in \mathbf{X}$ *să fie echivalente este ca matricea asociată lui* **X**, $\mathfrak{A}_X$, *să aibă toate elementele egale cu* 1.

Demonstrație. Fie $\mathfrak{A}_X = (\alpha_{ij})$, $i, j \in \{1, 2, \ldots, p\}$. Din $\alpha_{ij} = 1$ pentru orice $i, j \in \{1, 2, \ldots, p\}$ rezultă că stările $z_1, z_2, \ldots, z_p$ sînt compatibile două cîte două. Conform lemei din 8.6, relația de compatibilitate este reflexivă și simetrică. Pentru terminarea demonstrației rămîne să mai arătăm că relația dintre stările lui **X** este și tranzitivă. Din faptul că matricea $\mathfrak{A}_X$ are toate elementele egale cu 1, rezultă că pentru orice triplet i, j, k, dată cu $\alpha_{ij} = 1$ și $\alpha_{jk} = 1$, este egal cu 1 și α_{ik}, adică în submulțimea **X** se verifică și tranzivititatea, ceea ce demonstrează lema.

Fie $\mathfrak{A}_X$ matricea asociată submulțimii $\mathbf{X} \subset \mathbf{Z}$. Presupunem că $\mathfrak{A}_X$ are toate elementele egale cu 1.

Definiția 2. *O submatrice* $\mathfrak{A}_X$ a matricei $\mathfrak{A}$, care are toate elementele egale cu 1, se numește *maximală* în $\mathfrak{A}$, dacă nu există o altă matrice $\mathfrak{A}_Y$, $\mathbf{Y} \subset \mathbf{Z}$, cu toate elementele egale cu 1, astfel încît **X** să fie submulțimea strictă a lui **Y**. Submulțimea $\mathbf{X} \subset \mathbf{Z}$ corespunzătoare matricei $\mathfrak{A}_X$, maximală în $\mathfrak{A}$, se va numi *submulțime maximală în* **Z**.

Vom nota prin $\mathscr{F}(\mathfrak{A})$ familia matricelor maximale în $\mathfrak{A}$ și prin $\mathscr{Z}(\mathfrak{A})$ familia submulțimilor maximale în **Z**. Ținînd seama de lema 1, rezultă că restricția relației de compatibilitate la o submulțime maximală în **Z** este o relație de echivalență.

Să considerăm un automat complet **A** și să analizăm relația de compatibilitate a automatului **A**. Vom demonstra următoarea teoremă referitoare la relația de compatibilitate a automatului complet **A**.

Teorema 1. *Relația de compatibilitate* $\mathscr{R}(\mathbf{A})$ *a unui automat complet* **A** *este o relație de echivalență.*

Demonstrație. Întrucît relația de compatibilitate este reflexivă și simetrică, pentru demonstarea teoremei este suficient să arătăm că relația $\mathscr{R}(\mathbf{A})$ este și tranzitivă.

Vom face demonstrația prin absurd.

În acest scop presupunem că există trei stări z_i, z_j, z_k astfel încît $(z_i, z_j) \in \mathscr{R}(\mathbf{A})$, $(z_i, z_k) \in \mathscr{R}(\mathbf{A})$ dar $(z_j, z_k) \notin \mathscr{R}(\mathbf{A})$. Altfel spus, starea z_i este compatibilă cu z_j și cu z_k, dar stările z_j și z_k nu sînt compatibile între ele. Dacă $(z_j, z_k) \notin \mathscr{R}(\mathbf{A})$, există cel puțin o comandă x_1 și o pereche de stări (z_{j1}, z_{k1}) astfel încît $z_{j1} = \delta(z_j, x_1)$, $z_{k1} = \delta(z_k, x_1)$ și $(z_{j1}, z_{k1}) \notin \mathscr{R}(\mathbf{A})$, întrucît în caz contrar z_j și z_k ar fi compatibile. Dar atunci există o comandă x_2 și o pereche de stări $(z_{j2}, z_{k2}) \notin \mathscr{R}(\mathbf{A})$ astfel încît $z_{j2} = \delta(z_{j1}, x_2)$, $z_{k2} = \delta(z_{k1}, x_2)$. În general există un șir de comenzi $x_1, x_2, \ldots, x_p$ și un șir de perechi în stări incompatibile (z_{j0}, z_{k0}), (z_{j1}, z_{k1}), (z_{j2}, z_{k2}), $\ldots$, (z_{jp}, z_{kp}) în care o pereche se deduce din precedenta după regula $z_{js} = \delta(z_{js-1}, x_s)$, $z_{ks} = \delta(z_{ks-1}, x_s)$, $s \in \{1, 2, \ldots, p\}$, $z_{j0} = z_j$, $z_{k0} = z_k$.

Fie λ_{js} și λ_{ks} ieșirile corespunzătoare stărilor z_{js-1} și respectiv z_{ks-1}, sub comanda x_s: $\lambda_{js} = \lambda(z_{js-1}, x_s)$, $\lambda_{ks} = \lambda(z_{ks-1}, x_s)$. Corespunzător șirului de perechi de stări incompatibile vom obține un sir de perechi de ieșiri

$$(\lambda_{j1}, \lambda_{k1}), (\lambda_{j2}, \lambda_{k2}), \ldots, (\lambda_{jp}, \lambda_{kp}).$$

Evident nu putem presupune că $\lambda_{js} = \lambda_{ks}$, $(\forall)\, s \in \{1, 2, \ldots, p\}$, pentru orice șir de perechi de stări obținut din perechea (z_j, z_k) prin procedeul folosit, întrucît în acest caz stările z_j și z_k ar induce aceleași aplicații și conform definiției 2 din 8.4.2 ar fi echivalente. Există deci un s astfel încît $\lambda_{js} \neq \lambda_{ks}$, $s \in \{1, 2, \ldots, p\}$.

Construim șirurile de perechi de ieșiri:

$$(\lambda_{i1}, \lambda_{j1}), (\lambda_{i2}, \lambda_{j2}), \ldots, (\lambda_{is}, \lambda_{js}),$$

$$(\lambda_{i1}, \lambda_{k1}), (\lambda_{i2}, \lambda_{k2}), \ldots, (\lambda_{is}, \lambda_{ks})$$

și șirurile de perechi de stări

$$(z_{i1}, z_{k1}), (z_{i2}, z_{k2}), \ldots, (z_{is}, z_{ks})$$

în care $z_{it} = \delta(z_{it-1}, x_t)$, $t \in [1, s]$, $z_{i0} = z_i$ iar z_{jt}, z_{kt} au semnificațiile menționate. Se deduc imediat implicațiile :

$$(z_i, z_j) \in \mathscr{R}(\mathbf{A}) \Rightarrow (\lambda_{i1} = \lambda_{j1}, \lambda_{i2} = \lambda_{j2}, \ldots, \lambda_{is} = \lambda_{js})$$

și

$$(z_i, z_k) \in \mathscr{R}(\mathbf{A}) \Rightarrow (\lambda_{i1} = \lambda_{k1}, \lambda_{i2} = \lambda_{k2}, \ldots, \lambda_{is} = \lambda_{ks}),$$

de unde rezultă $\lambda_{js} = \lambda_{ks} = \lambda_{is}$, ceea ce contrazice presupunerea că $\lambda_{js} \neq \lambda_{ks}$, astfel că teorema este demonstrată.

După cum s-a arătat în cap. 1, unei relații de echivalență i se atașează biunivoc o partiție. Rezultă că relației de compatibilitate i se atașează o partiție a stărilor în automatele complete. Altfel spus, relația de compatibilitate realizează o clasificare a stărilor automatului complet **A** în clase de echivalență.

Vom extinde acest rezultat la automatele incomplete. Fie $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p\}$ o submulțime a familiei submulțimilor maximale în **Z**, $\mathbf{X}_i \in \mathbf{Z}(\mathfrak{A})$, $i \in \{1, 2, \ldots, p\}$. Vom spune că o submulțime $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p\}$ a familiei submulțimilor maximale în **Z** realizează o clasificare a stărilor automatului **A**, dacă submulțimile lui **X** acoperă mulțimea **Z** a stărilor și dacă nu există o submulțime strictă a lui **X** cu această proprietate.

Vom da un algoritm pentru determinarea familiei de submulțimi maximale ale mulțimii stărilor unui automat dat. Fie $\mathfrak{A}$ matricea de compatibilitate a automatului, $\mathfrak{A}^0$ matricea obținută din $\mathfrak{A}$ prin ștergerea liniilor și coloanelor care au 0 în linia și coloana întîi, și $\mathfrak{A}^1$ matricea obținută din $\mathfrak{A}$ prin ștergerea liniei și coloanei întîi.

Vom nota prin $\overline{\mathscr{Z}}(\mathfrak{A}^1) \subset \mathscr{Z}(\mathfrak{A}^1)$ familia de mulțimi maximale asociate lui $\mathfrak{A}^1$ și neincluse în mulțimile din $\mathscr{Z}(\mathfrak{A}^0)$.

Lema 2. *Dacă* $\mathscr{Z}(\mathfrak{A})$ *este familia de submulțimi maximale asociate lui* $\mathfrak{A}$, $\mathscr{Z}(\mathfrak{A}^0)$ *și* $\mathscr{Z}(\mathfrak{A}^1)$ *subfamiliile de submulțimi maximale asociate lui* $\mathfrak{A}^0$ *și respectiv* $\mathfrak{A}^1$, *atunci* $\mathscr{Z}(\mathfrak{A}) = \mathscr{Z}(\mathfrak{A}^0) \cup \overline{\mathscr{Z}}(\mathfrak{A}^1)$.

Demonstrație. $\mathscr{Z}(\mathfrak{A}^0)$ conține submulțimile maximale din $\mathscr{Z}(\mathfrak{A})$ care cuprind pe z_1 iar $\overline{\mathscr{Z}}(\mathfrak{A}^1)$ conține submulțimile maximale din $\mathscr{Z}(\mathfrak{A})$ care nu conțin pe z_1. Rezultă că $\mathscr{Z}(\mathfrak{A}^0) \cup \overline{\mathscr{Z}}(\mathfrak{A}^1)$ acoperă pe $\mathscr{Z}(\mathfrak{A})$. Pe de altă parte, orice mulțime maximală din $\mathscr{Z}(\mathfrak{A}^0)$ sau $\overline{\mathscr{Z}}(\mathfrak{A}^1)$ se include în $\mathscr{Z}(\mathfrak{A})$, deci există și incluziunea inversă, ceea ce demonstrează lema.

Plecînd de la aceste rezultate, se poate da algoritmul de clasificare a stărilor unui automat **A**. Aceasta revine la determinarea familiei de submulțimi maximale ale mulțimii stărilor sau la determinarea familiei de submatrice maximale ale lui $\mathfrak{A}$.

Se presupune cunoscuta matricea de compatibilitate $\mathfrak{A}$. Vom conveni să asociem unei submulțimi $\mathfrak{A}' \subset \mathfrak{A}$ două mulțimi : mulțimea stărilor asociate lui $\mathfrak{A}'$, fie **X** aceasta, și o mulțime, fie **Y**, care va fi construită în procesul de aplicare a algoritmului. Cînd dorim să punem în evidență cele două mulțimi, vom scrie $\mathfrak{A}_{X,Y}$. Pentru condensarea scrierii de obicei în locul matricei $\mathfrak{A}_{X,Y}$ vom scrie simplu $(\mathbf{X}, \mathbf{Y})$.

Pentru descrierea algoritmului mai sînt necesare două tabele T_1 și T_2 pentru reținerea unor matrice intermediare și un tabel T pentru înscrierea submulțimilor maximale în **Z**.

Descrierea algoritmului. 1. Se înscrie în tabelul T_1 matricea $\mathfrak{A}_{X,Y} = \mathfrak{A}$, unde $\mathbf{X} = \mathbf{Z}$, $\mathbf{Y} = \emptyset$.

2. Dacă există cel puțin o matrice $\mathfrak{A}_{X',Y'}$, în T_1, $\mathfrak{A}_{X,Y} = \mathfrak{A}_{X',Y'}$, se șterge $\mathbf{X}_{X',Y'}$ din T_1 și se trece la pasul 3 ; dacă nu, se trece la pasul 5.

3. Dacă matricea $\mathfrak{A}_{X,Y}$ are toate elementele egale cu 1, se trece la pasul 4 ; dacă nu, se trece la pasul 6.

4. Dacă submulțimea $\mathbf{X} \cup \mathbf{Y}$ se include într-o submulțime din T, se trece la pasul 2 ; dacă nu, se înscrie $\mathbf{X} \cup \mathbf{Y}$ în T și apoi se trece la pasul 2.

5. Dacă tabelul T_2 conține cel puțin o matrice, se mută matricele din T_2 în T_1, se golește tabelul T_2 și se trece la pasul 2 ; dacă nu, se trece la pasul 8.

6. Dacă matricea $\mathfrak{A}_{X,Y}$ are elementele primei linii egale cu 1, se înlocuiește matricea $\mathfrak{A}_{X,Y}$ cu matricea $\mathfrak{A}_{X',Y'}$, unde

$X' = X \setminus t$, $Y' = Y \cup t$, t fiind starea corespunzătoare primei linii și se trece la pasul 3; dacă nu, se trece la pasul 7.

7. Se inscriu în tabelul T_2 matricele $\mathfrak{A}^0_{X,Y}$ și $\mathfrak{A}^1_{X,Y}$ și se trece la pasul 2.

8. Tabelul T conțin familia de submulțimi de stări maximale în **Z**. Procesul de determinare a submulțimilor maximale în **Z** încetează.

Exemplul 1. Să se determine familia de submulțimi maximale în **Z** pentru automatul din tabelul 10.3.

Matricea de compatibilitate a fost determinată în 10.2 și este

$$\mathfrak{A} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Pentru condensarea scrierii vom conveni să notăm pașii în ordinea în care apar și rezultatele pentru fiecare pas, fără comentarii.

1. $T_1 = [(\{1, 2, 3, 4, 5, 6\}, \emptyset)]$.
2. $\mathfrak{A}_{X,Y} = \mathfrak{A}$, $\mathbf{X} = \{1, 2, 3, 4, 5, 6\}$, $\mathbf{Y} = \emptyset$, $T_1 = \emptyset$,
7. $T_2 = [(\{1, 2, 3, 4, 5\}, \emptyset), (\{2, 3, 4, 5, 6\}, \emptyset)]$.
5. $T_1 = T_2$, $T_2 = \emptyset$.
2. $\mathfrak{A}_{X,Y} = (\{1, 2, 3, 4, 5\}, \emptyset)$, $T_1 = [(\{2, 3, 4, 5, 6\}. \emptyset)]$.
6. $\mathfrak{A}_{X,Y} = (\{2, 3, 4, 5\}, \{1\})$.
7. $T_2 = [(\{2\}, \{1\}), (\{3, 4, 5\}, \{1\})]$.
2. $\mathfrak{A}_{X,Y} = (\{2, 3, 4, 5, 6\}, \emptyset)$, $T_1 = \emptyset$.
7. $T_2 = [(\{2\}, \{1\}), (\{3, 4, 5\}, \{1\}), (\{3, 4, 5, 6\}, \emptyset)]$.
5. $T_1 = T_2$, $T_2 = \emptyset$.
2. $\mathfrak{A}_{X,Y} = (\{2\}, \{1\})$, $T_1 = [(\{3, 4, 5\}, \{1\}), (\{3, 4, 5, 6\}, \emptyset)]$.
4. $T = [\{1, 2\}]$.
2. $\mathfrak{A}_{X,Y} = (\{3, 4, 5\}, \{1\})$, $T_1 = [(\{3, 4, 5, 6\}, \emptyset]$.
7. $T_2 = [(\{3, 5\}, \{1\}), (\{4, 5\}, \{1\})]$.
2. $\mathfrak{A}_{X,Y} = (\{3, 4, 5, 6\}, \emptyset)$, $T_1 = \emptyset$.

7. $T_2 = [(\{3, 5\}, \{1\}), (\{4, 5\}, \{1\}), (\{3, 5, 6\}, \emptyset), (\{4, 5, 6\}, \emptyset)]$.
5. $T_1 = T_2$, $T_2 = \emptyset$.
2. $\mathfrak{A}_{X,Y} = (\{3, 5\}, \{1\}]$, $T_1 = [(\{4, 5\}, \{1\}), (\{3, 5, 6\}, \emptyset)$. $(\{4, 5, 6\}, \emptyset)]$.
4. $T = [\{1, 2\}, \{1, 3, 5\}]$.
2. $\mathfrak{A}_{X,Y} = (\{4, 5\}, \{1\})$, $T_1 = [(\{3, 5, 6\}, \emptyset), (\{4, 5, 6\}, \emptyset)]$.
4. $T = [\{1, 2\}, \{1, 3, 5\}, \{1, 4, 5\}]$.
2. $\mathfrak{A}_{X,Y} = (\{3, 5, 6\}, \emptyset)$, $T_1 = [(\{4, 5, 6\}, \emptyset)]$.
6. $\mathfrak{A}_{X,Y} = (\{5, 6\}, \{3\})$.
7. $T_2 = [\{5\}, \{3\}), (\{6\}, \{3\})]$.
2. $\mathfrak{A}_{X,Y} = (\{4, 5, 6\}, \emptyset)$, $T_1 = \emptyset$.
7. $T_2 = [(\{5\}, \{3\}), (\{6\}, \{3\}), (\{4, 5\}, \emptyset), (\{5, 6\}, \emptyset)]$.
5. $T_1 = T_2$, $T_2 = \emptyset$.
2. $\mathfrak{A}_{X,Y} = (\{5\}, \{3\})$, $T_1 = [(\{6\}, \{3\}), (\{4, 5\}, \emptyset), (\{5, 6\}, \emptyset)]$.
2. $\mathfrak{A}_{X,Y} = (\{6\}, \{3\})$, $T_1 = [(\{4, 5\}, \emptyset), (\{5, 6\}, \emptyset)]$.
4. $T = [\{1, 2\}, \{1, 3, 5\}, \{1, 4, 5\}, \{3, 6\}]$.
2. $\mathfrak{A}_{X,Y} = (\{4, 5\}, \emptyset)$, $T_1 = [(\{5, 6\}, \emptyset)]$.
2. $\mathfrak{A}_{X,Y} = (\{5, 6\}, \emptyset)$, $T_1 = \emptyset$.
7. $T_2 = [(\{5\}, \emptyset), (\{6\}, \emptyset)]$.
5. $T_1 = T_2$, $T_2 = \emptyset$.
2. $\mathfrak{A}_{X,Y} = (\{5\}, \emptyset)$, $T_1 = [(\{6\}, \emptyset)]$.
2. $\mathfrak{A}_{X,Y} = (\{6\}, \emptyset)$, $T_1 = \emptyset$.

8. Familia de submulțimi maximale în **Z** este

$$Z(\mathfrak{A}) = \{\{1, 2\}, \{1, 3, 5\}, \{1, 4, 5\}, \{3, 6\}\}.$$

Procesul încetează.

Observație. Nu sînt trecuți pașii care au constituit puncte de decizie intermediare. De exemplu, după prima apariție a pasului 2 s-a trecut la pasul 3 și deoarece $\mathfrak{A}_{X,Y}$ nu a avut toate elementele egale cu 1, s-a trecut la pasul 6. De la pasul 6 s-a trecut la pasul 7, deoarece $\mathfrak{A}_{X,Y}$ nu a avut nici elementele primei linii egale cu 1. Pașii 3 și 6 nu au fost trecuți, întrucît ei nu au condus la schimbarea stării tabelelor T, T_1, T_2 sau a matricei $\mathfrak{A}_{X,Y}$.

Dacă familia $Z(\mathfrak{A})$ a submulțimilor maximale în $\mathbf{Z}$ nu este o acoperire minimală a lui $\mathbf{Z}$, este necesar să determinăm aceste acoperiri minimale. În acest scop vom folosi algoritmul dat în cap. 6.

Exemplul 2. Să se determine acoperirile minimale ale lui $\mathbf{Z}$ pentru automatul cu familia de submulțimi maximale în $\mathbf{Z}$ din exemplul 1.

Vom asocia mulțimilor din $\mathscr{Z}(\mathfrak{A})$ variabilele booleene

$$A \leftrightarrow \{1, 2\},\ B \leftrightarrow \{1, 3, 5\},\ C \leftrightarrow \{1, 4, 5\},\ D \leftrightarrow \{3, 6\}.$$

Expresia booleană asociată lui $\mathscr{Z}(\mathfrak{A})$ va fi

$$(A + B + C)\,A\,(B + D)\,C\,(B + C)\,D = ACD.$$

Există deci o singură acoperire minimală formată din mulțimile

$$\mathbf{Z}_1 = \{1, 2\},\ \mathbf{Z}_2 = \{1, 4, 5\} \text{ și } \mathbf{Z}_3 = \{3, 6\}.$$

10.4. Definirea funcțiilor de tranziție și de ieșire ale automatului redus

Dacă nu există $\mathbf{Z}_j \in \mathbf{Z}$, astfel încît $\delta(\mathbf{Z}_i, x) \in \mathbf{Z}_j$, atunci se analizează dacă nu există în $\mathbf{Z}_i$ stări redondante, adică stări care apar și în alte submulțimi din $\mathbf{Z}$. Se caută ca prin eliminarea stărilor redondante din $\mathbf{Z}_i$ să se ajungă la verificarea condiției 3.

Exemplu. Să se determine funcțiile de tranziție și de ieșire pentru automatul redus din exemplele 1 și 2 din 10.2.

Stările automatului redus sînt $\mathbf{Z}_1 = \{1, 2\}$, $\mathbf{Z}_2 = \{1, 4, 5\}$ și $\mathbf{Z}_3 = \{3, 6\}$ Funcția de tranziție este următoarea :

$$\delta^*(\mathbf{Z}_1, x^1) = \{3\}, \qquad \delta^*(\mathbf{Z}_1, x^2) = \{3, 5\},$$

$$\delta^*(\mathbf{Z}_2, x^1) = \{5\}, \qquad \delta^*(\mathbf{Z}_2, x^2) = \{3, 6\},$$

$$\delta^*(\mathbf{Z}_3, x^1) = \{4\}, \qquad \delta^*(\mathbf{Z}_3, x^2) = \emptyset.$$

Se observă că $\delta^*(\mathbf{Z}_1, x^2)$ nu este inclus în nici una dintre submulțimile $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{Z}_3$. Dacă înlăturăm starea 1 din $\mathbf{Z}_1$ care este redontantă, se obține noua stare a automatului redus $\mathbf{Z}_1' = \{2\}$, care înlocuiește pe $\mathbf{Z}_1$ și

$\delta^*(\mathbf{Z}_1', x^2) = \{5\} \subset \mathbf{Z}_2$. Se obține automatul redus din tabelul 10.4, unde $\mathbf{K} = \mathbf{Z}_1'$, $\mathbf{C} = \mathbf{Z}_2$, $\mathbf{D} = \mathbf{Z}_3$. Funcția de ieșire a automatului este următoarea :

$$\lambda^*(\mathbf{Z}_1', x^1) = \lambda(2, x^1) = 0, \qquad \lambda^*(\mathbf{Z}_1', x^2) = \lambda(2, x^2) = 0,$$

$$\lambda^*(\mathbf{Z}_2, x^1) = \lambda(5, x^1) = 1, \qquad \lambda^*(\mathbf{Z}_2, x^2) = \lambda(5, x^2) = 0,$$

$$\lambda^*(\mathbf{Z}_3, x^1) = \lambda(3, x^1) = 0, \qquad \lambda^*(\mathbf{Z}_3, x^2) = \lambda(6, x^2) = 1.$$

Se vede încă o dată că dacă se pleacă de la mulțimea redusă de submlțimi de stări echivalente, nu se ajunge la automatul cu număr minim de stări căutat.

Să descriem complet algoritmul de reducere a automatelor la automate cu număr minim de stări.

Pasul 1. Se scrie funcția $\mathscr{E}(\mathbf{A})$ și din aceasta se deduce implicantul prim generator al automatului redus.

Pasul 2. Se asociază implicantului prim generator matricea $\mathfrak{A}$.

Pasul 3. Se construiesc mulțimile de submulțimi maximale de stări echivalente ale lui $\mathbf{Z}$, folosind algoritmul de căutare descris mai înainte.

Pasul 4. Se caută acoperirile minimale ale mulțimii stărilor.

Pasul 5. Se aleg dintre acoperirile minimale cele care îndeplinesc condiția 3, referitoare la funcțiile de tranziție sau se realizează acest lucru prin eliminarea din stările reduse a unor stări inițiale redondante.

Pasul 6. Se calculează funcțiile de tranziție și de ieșiri.

Rezultatele obținute s-au referit la automate incomplete. În continuare vom demonstra că relația de compatibilitate a stărilor în automatele incomplete se reduce la relația de echivalență dacă se aplică la automatele complete.

Lemă. *Condiția necesară și suficientă pentru ca două stări p și q ale automatului complet $\mathbf{A}$ să fie echivalente este ca e_{pq} să apară nenegată în elementul generator al automatului redus $\mathscr{E}_A$.*

Demonstrație. Dacă e_{pq} apare nenegată în $\mathscr{E}_A$, atunci stările p și q sînt compatibile. Rămîne să arătăm că se verifică și tranzitivitatea. Este suficient să arătăm că dacă e_{pq} și e_{qr} sînt nenegate în $\mathscr{E}_A$, atunci și e_{pr} este nene-

gată. Presupunem contrariul. Atunci există un cuvînt de intrare $J = x^1x^2 \ldots x^k$ pentru care :

$$\lambda(p_i, x^i) = \lambda(r_i, x^i), \quad i = 1, 2, \ldots, k-1,$$

$$\lambda(p_k, x^k) \neq \lambda(r_k, x^k).$$

Automatul fiind complet definit, cuvîntul J se poate aplica și stării q și datorită presupunerii vom avea

$$\lambda(p_i, x^i) = \lambda(q_i, x^i), \quad i = 1, 2, \ldots, k,$$

$$\lambda(q_i, x^i) = \lambda(r_i, x^i),$$

ceea ce contrazice relațiile precedente, deci presupunerea este absurdă.

ASIGNAREA VARIABILELOR INTERMEDIARE

11.1. Definiții

Un automat cu N stări poate fi realizat cu M elemente binare, unde M este dat de relația (v. 9.5)

$$2^{M-1} < N \leqslant 2^M.$$

Dacă din punct de vedere teoretic orice combinație a celor M elemente poate fi asignată oricărei stări a automatului, din punct de vedere practic, numărul acestor asignări este limitat de unele condiții de realizabilitate și de condiții de bună funcționare, printre care sînt condiția tehnică de bună funcționare și eliminarea hazardului.

În acest capitol ne propunem să dăm o metodă de asignare a variabilelor secundare care respectă condiția tehnică de bună funcționare, urmînd ca în capitolul următor să tratăm problema hazardului.

Definiție. Numim *stare* a variabilelor de intrare orice *combinație de valori* a acestor variabile. În același sens putem vorbi de *stările variabilelor secundare* și *stările variabilelor de ieșire.*

Nu trebuie confundată noțiunea de *combinație a variabilelor secundare* cu noțiunea de *stare a automatului*, întrucît pentru un automat dat nu orice combinație a elementelor intermediare este și o stare a automatului. Stă-

rile automatului se găsesc, în general, într-o submulțime a mulțimii tuturor combinațiilor elementelor intermediare. Este adevărat că se poate stabili o corespondență între o submulțime a mulțimii combinațiilor variabilelor intermediare și mulțimea stărilor automatului, în care unei stări date a automatului să corespundă o combinație dată a variabilelor secundare, după cum se va vedea.

Cu această precizare se evită confuzia între noțiunea de combinație a variabilelor și noțiunea de stare a automatului.

O stare a variabilelor de intrare este un vector cu n componente, unde n este numărul de variabile de intrare. La fel, combinațiile de variabile intermediare și de variabile de ieșire sînt vectori.

Să considerăm mai întîi cazul binar. Fie $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ un vector binar, $\alpha_i = 0$ sau 1 pentru orice $i = 1, 2, \ldots, n$. După cum se știe, doi vectori n dimensionali binari α și β sînt vecini dacă vectorul

$$\alpha \oplus \beta = (\alpha_1 \oplus \beta_1, \alpha_2 \oplus \beta_2, \ldots, \alpha_n \oplus \beta_n),$$

unde $\alpha_i \oplus \beta_i$ este suma modulo 2 a componentelor α_i și β_i, îndeplinește condiția $\alpha_i \oplus \beta_i = 0$ pentru orice i, în afară de unul singur, pentru care $\alpha_i \oplus \beta_i = 1$.

Definiție. Spunem că un automat *respectă condiția tehnică de bună funcționare* dacă trecerea de la o stare a variabilelor de intrare la alta conduce la trecerea automatului într-o stare vecină sau la trecerea automatului de la o stare la alta printr-un șir de stări vecine, două cîte două.

Exemplu. Automatul cu două semnale de intrare x_1 și x_2 și trei elemente intermediare y_1, y_2, y_3, a cărui matrice de tranziție a stărilor este dată în tabelul 11.1, respectă condiția tehnică de bună funcționare. Într-adevăr, se poate verifica ușor că trecerile între stări se fac numai prin stări vecine. De exemplu, starea 110 a automatului sub comanda 11 a semnalelor de intrare este stabilă. Dacă se dă comanda vecină 10, automatul trece în starea 100 care este vecină cu 110. Starea 000 sub comanda 00 este însă o stare stabilă a automatului. În acest caz, dacă se dă comanda vecină 01, se trece mai întîi în starea vecină 001, apoi din aceasta se trece fără să schimbăm comanda în starea 101 și mai departe în starea 111 care este o stare stabilă. Are loc un proces autonom de trecere dintr-o stare stabilă în altă stare stabilă prin intermediul unor stări stabile, vecine două cîte două.

Tabelul 11.1

$y_1y_2y_3$ \ x_1x_2	00	01	11	10
000	000	001	000	001
001	101	101	—	101
101	111	111	—	111
111	101	111	110	111
110	100	100	110	100
100	100	100	000	100

11.2. Spațiul vectorial asociat mulțimii stărilor automatului. Variabile intermediare asociate

Fie $\mathcal{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ un automat finit cu matricea de tranziție a stărilor dată în tabelul 11.2, unde $\delta_{ij} = \delta(z_j, x_i)$, $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, p\}$.

Tabelul 11.2

z \ x	x_1	x_2	...	x_p
z_1	δ_{11}	δ_{12}	...	δ_{1p}
z_2	δ_{21}	δ_{22}	...	δ_{2p}
⋮	⋮	⋮		⋮
z_n	δ_{n1}	δ_{n2}	...	δ_{np}

Fie $(\mathbf{Z}, \oplus)$ mulțimea elementelor obținute din $\mathbf{Z} = \{z_1, z_2, \ldots, z_n\}$ prin operația $\oplus$, cu proprietățile:

$z_i \oplus z_j = z_j \oplus z_i$ (comutativitate),
$(z_i \oplus z_j) \oplus z_k = z_i \oplus (z_j \oplus z_k)$ (asociativitate),
$z_i \oplus \theta_0 = z_i$ (existența elementului unitate),
$z_i \oplus z_i = \theta_0$ (existența inversului),

adevărate pentru orice $i, j, k \in \{1, 2, \ldots, n\}$.

Proprietate. $z_i \oplus z_j = z_k \Rightarrow (z_i \oplus z_k = z_j$ și $z_j \oplus z_k = z_i)$.

Mulțimea $(\mathbf{Z}, \oplus)$ este un grup abelian în care fiecare element este propriul său invers, $z_i^{-1} = z_i$, iar elementul

unitate este rezultatul compunerii lui z_i cu el însuși, $\theta_0 = z_i \oplus z_i$, $(\forall)\, i \in \{1, 2, \ldots, n\}$.

Fie $\mathbf{K}$ mulțimea perechilor (i, j), $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, p\}$, pentru care starea δ_{ij} este definită și $\Theta = \{z_i \oplus \delta_{ij} \mid (i, j) \in \mathbf{K}\}$ este submulțimea grupului $(\mathbf{Z}, \oplus)$ care conține elementele obținute prin compunerea stărilor z_i, $i \in \{1, 2, \ldots, n\}$, cu fiecare dintre stările δ_{ij}, $j \in \{1, 2, \ldots, p\}$, în care poate trece direct sub una din comenzile date.

Fie $\theta_{ij} = z_i \oplus \delta_{ij}$, $\quad i, j \in \mathbf{K}$; $\quad \theta_{ij} \in \Theta \subset (\mathbf{Z}, \oplus)$.

Definiția 1. Relația $\theta_{ij} = z_i \oplus \delta_{ij}$ va fi numită *relația de trecere* de la starea z_i la starea δ_{ij}. Mulțimea relațiilor de trecere va fi notată prin $\mathscr{R}(\mathscr{A})$.

Presupunem că Θ conține $r + 1$ elemente, $\Theta = \{\theta_0, \theta_1, \ldots, \theta_r\}$, unde θ_0 este elementul unitate din $(\mathbf{Z}, \oplus)$.

Observație. Includerea lui θ_0 în Θ este naturală. Ea corespunde ipotezei că automatul conține cel puțin o stare stabilă.

În general, nu toate elementele lui Θ sînt liniar independente. Între aceste elemente pot exista relații rezultate din definiția lor sau relații existente între stările automatului care se transmit elementelor lui Θ.

Vom nota prin $\mathscr{R}(\Theta)$ mulțimea relațiilor care există între elementele lui Θ. Relațiile din $\mathscr{R}(\Theta)$ constituie un sistem de ecuații în care variabilele sînt elementele din Θ.

Definiția 2. Mulțimea Θ va fi numită *mulțimea elementelor de trecere* dintre stări, iar $\mathscr{R}(\Theta)$ *mulțimea relațiilor* dintre elementele de trecere.

Ținînd seama de proprietățile operației $\oplus$, rezultă că orice relație din $\mathscr{R}(\Theta)$ este o relație liniară. Altfel spus, $\mathscr{R}(\Theta)$ este un sistem de ecuații liniare, ale cărui variabile sînt elementele din Θ.

Dacă p este numărul de ecuații liniar independente din $\mathscr{R}(\Theta)$, atunci p variabile din Θ pot fi determinate prin rezolvarea sistemului $\mathscr{R}(\Theta)$, iar restul de $r - p = m$ pot fi luate arbitrar.

Fie $\Theta_0 = \{\theta_1, \theta_2, \ldots, \theta_m\}$ o mulțime de elemente din Θ, a căror valoare poate fi luată arbitrar, deci care sînt

liniar independente, m fiind numărul maxim de elemente liniar independente din Θ.

Teoremă. *Mulțimea stărilor unui automat* $\mathscr{A}(\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \delta, \lambda)$ *admite o reprezentare în spațiul vectorial* $\mathbf{B}_2^m$, *pe care îl vom numi spațiul vectorial asociat automatului, m fiind numărul maxim de elemente liniar independente din* Θ.

Demonstrație. Fie Θ_0 un sistem liniar independent de elemente din Θ și $\mathscr{R}(\Theta)$ sistemul de ecuații liniar independente care leagă elementele lui Θ. Atunci prin rezolvarea sistemului $\mathscr{R}(\Theta)$ orice element $\theta \in \Theta \setminus \Theta_0$ poate fi scris ca o combinație liniară în funcție de elementele lui Θ_0, fie

$$\theta = \bigoplus_{i=1}^{m} \mu_i \theta_i, \quad \mu_i = 0 \text{ sau } 1.$$

Convenim să asociem elementelor $\theta_i \in \Theta_0$ vectori binari, m-dimensionali

$$\theta_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{im})$$

cu proprietatea

$$i \neq k \Rightarrow \alpha_{ik} = 0,$$

$$i = k \Rightarrow \alpha_{ik} = 1, \quad (\forall)\, k \in \{1, 2, \ldots, m\}.$$

Atunci elementele $\theta_i \in \Theta_0$ constituie o bază în spațiul vectorial $\mathbf{B}_2^m$, iar elementele $\theta \in \Theta \setminus \Theta_0$ sînt reprezentate în $\mathbf{B}_2^m$ prin punctele

$$\theta = (\mu_1, \mu_2, \ldots, \mu_m),$$

ceea ce trebuia demonstrat.

Convenim să asociem fiecărei componente a vectorilor $\theta \in \Theta$ cîte o variabilă booleană u_j, $j \in \{1, 2, \ldots, m\}$, care ia valoarea μ_j pentru $\theta = (\mu_1, \mu_2, \ldots, \mu_m)$. Se poate scrie $u_j(\theta) = \mu_j$, $j \in \{1, 2, \ldots, m\}$, $(\forall)\, \theta \in \Theta$.

Definiția 2. Variabilele μ_j astfel asociate vor fi numite *variabilele intermediare ale automatului.* Operația de asociere de variabile booleene va fi numită *operația de asignare a elementelor intermediare sau secundare.*

Definiția 3. Două stări z_i și z_j se numesc *vecine* dacă $z_i \oplus z_j = \theta$, $\theta \in \Theta_0$.

Altfel spus, două stări sînt vecine dacă prin compunerea lor se obține un vector care are o singură componentă nenulă.

Exemplu. Fie automatul cu matricea de tranziție a stărilor din tabelul 11.3. Să se determine mulțimea elementelor de trecere și mulțimea relațiilor de trecere.

Tabelul 11.3

z \ x	00	01	11	10
z_1	z_1	z_3	z_4	z_1
z_2	z_1	z_4	z_2	z_2
z_3	z_4	z_3	z_2	z_1
z_4	z_4	z_4	z_4	z_2

După cum se poate verifica ușor, există șase elemente de trecere diferite de θ_0, $\theta_1 = z_1 \oplus z_3$, $\theta_2 = z_1 \oplus z_4$, $\theta_3 = z_1 \oplus z_2$, $\theta_4 = z_2 \oplus z_4$, $\theta_5 = z_3 \oplus z_4$ și $\theta_6 = z_2 \oplus z_3$, deci $\Theta = \{\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$. Prin eliminarea stărilor z_1, z_2, z_3, z_4 se obțin relațiile $\mathscr{R}(\Theta) = \{\theta_1\theta_2\theta_5 = \theta_0,\ \theta_1\theta_3\theta_6 = \theta_0,\ \theta_2\theta_3\theta_4 = \theta_0,\ \theta_4\theta_5\theta_6 = \theta_0\}$.

Observație. Pentru simplificare am convenit să omitem în scriere semnul $\oplus$. De exemplu scriem $\theta_1\theta_2\theta_5 = \theta_0$ în loc de $\theta_1 \oplus \theta_2 \oplus \theta_5 = \theta_0$ etc. Această convenție o vom păstra și în continuare cînd nu se ivesc confuzii.

Un sistem liniar independent de ecuații din $\mathscr{R}(\theta)$ este

$$\mathscr{R}_0(\Theta) = \{\theta_1\theta_2\theta_5 = \theta_0,\ \theta_1\theta_3\theta_6 = \theta_0,\ \theta_2\theta_3\theta_4 = \theta_0\}.$$

Prin rezolvarea sistemului $\mathscr{R}_0(\Theta)$ se pot determina toate elementele de trecere în funcție de cele liniar independente și deci se pot scrie relațiile de trecere ale automatului $\mathscr{R}(\mathscr{A})$ în funcție de aceste elemente. Există deci posibilitatea de a reprezenta mulțimea stărilor automatului în spațiul $\mathbf{B}_2^3$, luînd de exemplu ca sistem independent de elemente $\Theta_0 = \{\theta_1, \theta_2, \theta_3\}$ și rezolvînd sistemul $\mathscr{R}_0(\Theta)$ în raport cu θ_4, θ_5, $\theta_6 \in \Theta \setminus \Theta_0$. Se obține $\theta_4 = \theta_2\theta_3$, $\theta_5 = \theta_1\theta_2$ și $\theta_6 = \theta_1\theta_3$ sau sub formă vectorială: $\theta_4 = (0, 1, 1)$, $\theta_5 = (1, 1, 0)$ și $\theta_6 = (1, 0, 1)$.

În realitate, se poate realiza o reprezentare a mulțimii stărilor și în $\mathbf{B}_2^2$, dar în acest caz între stările automatului trebuie să existe relația $z_1z_2z_3z_4 = \theta_0$ care conduce la noi

relații între elementele lui Θ. După completarea cu aceste noi relații $\mathscr{R}(\Theta)$ devine

$$\mathscr{R}(\Theta) = \{\theta_1\theta_2\theta_5 = \theta_0,\ \theta_1\theta_3\theta_6 = \theta_0,\ \theta_2\theta_3\theta_4 = \theta_0,\ \theta_4\theta_5\theta_6 = \theta_0,\ \theta_1\theta_4 = \theta_0,\ \theta_2\theta_6 = \theta_0,\ \theta_1\theta_5 = \theta_0\}.$$

Un sistem liniar independent dintre relațiile $\mathscr{R}(\Theta)$ este

$$\mathscr{R}_0(\Theta) = \{\theta_1\theta_4 = \theta_0,\ \theta_2\theta_6 = \theta_0,\ \theta_3\theta_5 = \theta_0,\ \theta_1\theta_2\theta_3 = \theta_0\}.$$

Luînd arbitrare elementele θ_1, θ_2, se obține o reprezentare în $\mathbf{B}_2^2$: $\theta_3 = \theta_1\theta_2$, $\theta_4 = \theta_1$, $\theta_5 = \theta_1\theta_2$ și $\theta_6 = \theta_2$ sau sub formă vectorială $\theta_1 = \theta_4 = (1, 0)$, $\theta_2 = \theta_6 = (0, 1)$, $\theta_3 = \theta_5 = (1, 1)$.

Considerînd două variabile intermediare u_1 și u_2 și asociind de exemplu stării z_1 starea 00 a variabilelor u_1 și u_2, se obține matricea de tranziție a stărilor din tabelul 11.4, în care s-a considerat $z_2 = z_1 \oplus \theta_3 = 11$, $z_3 = z_1 \oplus \theta_1 = 10$ și $z_4 = z_1 \oplus \theta_2 = 0$.

Tabelul 11.4

z	u_1	u_2 \ x	00	01	11	10
z_1	0	0	00	10	01	00
z_2	1	1	00	01	11	11
z_3	1	0	01	10	11	00
z_4	0	1	01	01	01	11

11.3. Asignarea elementelor intermediare, cu respectarea condiției tehnice de bună funcționare

Metoda de asignare a elementelor intermediare nu conduce întotdeauna la asignări care respectă condiția tehnică de bună funcționare, după cum se poate verifica și pe exemplul din tabelul 11.4. Într-adevăr, trecerea din starea z_2 în starea z_1 sub comanda 00 se face prin variația simultană a celor două variabile u_1 și u_2, ceea ce poate conduce la hazard, după cum se va arăta mai departe.

Vom da o metodă de asignare a elementelor intermediare cu respectarea condiției tehnice de bună funcțio-

nare. Fie $z_i \oplus z_j = \overset{m}{\underset{k=1}{\Xi}} \mu_k \theta_k = 0$ una dintre relațiile de trecere $\mathscr{R}(\mathscr{A})$ ale automatului și $\mathbf{M}_i(\theta) = \{z_i \oplus \theta_k | \mu_k = 1, k \in \{1, 2, \ldots, m\}\}$, $\mathbf{M}_j(\theta) = \{z_j \oplus \theta_k | \mu_k = 1, k \in \{1, 2, \ldots, m\}\}$. Întrucît $\theta_k \in \Theta_0$, $\mathbf{M}_i(\theta)$ și $\mathbf{M}_j(\theta)$ reprezintă mulțimile stărilor variabilelor intermediare, vecine cu z_i și respectiv z_j. Există posibilitatea ca unele dintre stările lui $\mathbf{M}_i$ sau $\mathbf{M}_j$ să fie stări ale automatului.

Vom elimina din $\mathbf{M}_i$ și $\mathbf{M}_j$ toate stările care sînt stări ale automatului. Fie $\mathbf{M}_i^0(\theta) = \{z_i \oplus \theta_k | \mu_k = 1, k \in \mathbf{K}_i\}$ și $\mathbf{M}_j^0(\theta) = \{z_j \oplus \theta_k | \mu_k = 1, k \in \mathbf{K}_j\}$, unde $\mathbf{K}_i = \{k \in \{1, 2, \ldots, m\} | \mu_k = 1, z_i \oplus \theta_k \in \mathbf{Z}\}$, iar $\mathbf{Z}$ este mulțimea de stări ale automatului.

Vom nota prin $\mathbf{M}_0 = \bigcup (\mathbf{M}_i^0(\theta), \mathbf{M}_j^0(\theta))$, $\theta \in \Theta \setminus \Theta_0$, și prin $\mathbf{S}_0 = \{z_i | \exists\, z_i \oplus \theta \in \mathbf{M}_0\}$. Mulțimea $\mathbf{M}_0$ conține stări ale elementelor intermediare care nu sînt stări ale automatului iar $\mathbf{S}_0$ stările automatului care intervin în difinirea elementelor din $\mathbf{M}_0$.

Pentru fiecare $z \in \mathbf{S}_0$ vom introduce cîte o stare suplimentară $\bar{z} = z \oplus \theta$, unde $z \oplus \theta$ este unul dintre elementele lui $\mathbf{M}_0$ în care intervine z.

Fie $d(\theta) = \sum_{j \in \{1,2,\ldots,m\}} u_j(\theta)$ distanța dintre elementele θ și θ_0.

Definiția 2. Distanța $d = \max_{\theta \in \Theta} d(\theta)$ va fi numită *diametrul* mulțimii stărilor *automatului* sau, simplu, *diametrul automatului* și reprezintă distanța maximă dintre stările automatului între care există tranziție.

Lema 1. *Adăugarea de stări suplimentare unui automat oarecare* **A** *conduce la reducerea diametrului acestuia.*

Demonstrație. Fie $\theta = \theta_1 \theta_2 \ldots \theta_d$, $\theta_i \in \Theta_0$, $(\forall) i \in \{1, 2, \ldots, d\}$, elementul de trecere între două stări z_i și z_j, între care distanța este egală cu diametrul automatului și x_k o comandă sub care starea z_i trece în starea z_j.

Considerăm un element oarecare al mulțimii $\{\theta_1, \theta_2, \ldots, \theta_d\}$, fie pentru simplificarea scrierii θ_1. Dacă starea $z_i \oplus \theta_1 = z_j \oplus \theta_2 \theta_3 \ldots \theta_r = z_a \in \mathbf{Z}$ și dacă sub comanda x_k starea z_a trece în starea z_j sau starea $\delta(z_a, x_k)$ este nedefinită se poate înlocui tranziția $z_i \to z_j$ cu tranziția $z_i \to z_a \to z_j$, funcționarea automatului rămînînd neschimbată, însă distanța maximă dintre stări scade cu 1. Dacă se procedează la fel cu toate perechile de stări de distanță maximă, diametrul automatului se reduce la rîndul său cu 1. Dacă $z_a \notin \mathbf{Z}$, este suficient să adăugăm starea z_a la stările automatului și să procedăm ca mai înainte. Cu aceasta lema este demonstrată.

Realizăm o partiție a lui $\mathbf{M}_0$ astfel: notăm prin $\mathbf{M}_0^i$ mulțimea stărilor din $\mathbf{M}_0$ în care apare z_i · $\mathbf{M}_0^i = \emptyset$, dacă $z_i \notin \mathbf{S}_0$. Atunci $\mathbf{M}_0^i = \bigcup_{k \in \mathbf{N}_i} (z_i \oplus \theta_k)$, unde $\mathbf{N}_i \in \{1, 2, \ldots, m\}$ este mulțimea indicilor elementelor $\theta_k \in \Theta_0$ care se compun cu z_i în definirea elementelor lui $\mathbf{M}_0$. Rezultă $\mathbf{M}_0 = \bigcup_{i \in \{1,2,\ldots,m\}} \mathbf{M}_0^i$, $\mathbf{M}_0^i \cap \mathbf{M}_0^j = \emptyset$, $(\forall)\ i, j \in \{1, 2, \ldots, m\}$, $i \neq j$.

Lema 2. *Implicanții primi ai funcției* $f = \prod_{z_i \in \mathbf{S}_0} \sum_{k \in \mathbf{N}_i} (z_i \oplus \theta_k)$ *ne dau mulțimea posibilităților de adăugare de stări suplimentare.*

Demonstrația rezultă din faptul că fiecare implicant prim al lui f conține cîte o stare din $\mathbf{M}_0$ pentru fiecare z_i și numai una singură.

Lemele 1 și 2 ne pun la dispoziție o metodă de a reduce diametrul unui automat dat și de a-l transforma într-un automat, care respectă condiția tehnică de bună funcționare.

Elementele prezentate pînă acum ne permit să formulăm un algoritm de transformare a unui automat dat într-un automat care respectă condiția tehnică de bună funcționare.

Descrierea algoritmului. 1. Se determină mulțimea elementelor de trecere $\Theta = \{z_i \oplus \delta_{ij} | i, j \in \mathbf{K}\}$, unde $\mathbf{K}$ este mulțimea perechilor (i, j), $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, p\}$, pentru care starea $\delta_{ij} = \delta(z_i, z_j)$ este definită.

2. Se determină mulțimea relațiilor $\mathscr{R}(\Theta)$ dintre elementele de trecere. În $\mathscr{R}(\Theta)$ se includ atît relațiile rezultate din definiția elementelor $\theta \in \Theta$ cît și cele induse de relațiile existente între stările automatului.

3. Se rezolvă sistemul de ecuații $\mathscr{R}(\Theta)$. Dacă r este numărul de elemente din Θ și p numărul de relații din $\mathscr{R}(\Theta)$, atunci $m = r - p$ elemente din Θ se dau arbitrar iar p dintre ele se determină prin rezolvarea sistemului de ecuații. Fie Θ_0 mulțimea elementelor din Θ luate arbitrar. Se scriu toate elementele din Θ în funcție de elementele din Θ_0.

4. Se scrie matricea de trecere obținută din matricea de tranziție a stărilor prin înlocuirea stărilor cu elementele de trecere corespunzătoare.

5. Se verifică dacă automatul satisface condiția tehnică de bună funcționare. Dacă da, procesul este terminat. Dacă nu, se trece la punctul 6.

6. Se determină mulțimile

$$\mathbf{M}_i^0(\theta)=\{z_i \oplus \theta_k \mid \mu_k=1,\ k\in\mathbf{K}_i\} \text{ și } \mathbf{M}_j^0(\theta)=\{z_j \oplus \theta_k \mid \mu_k=1,\ k\in\mathbf{K}_j\}$$

de stări ale variabilelor intermediare vecine cu z_i și respectiv cu z_j, pentru orice pereche de stări $(z_i,\ z_j)$ pentru care $z_i \oplus z_j \in \Theta \setminus \Theta_0$ și $\mathbf{M}_0 = \bigcup_{\theta} (\mathbf{M}_i^0(\theta) \cup \mathbf{M}_j^0(\theta))$, $\theta \in \Theta \setminus \Theta_0$.

7. Se determină mulțimea $\mathbf{S}_0$ a stărilor automatului care intervin în $\mathbf{M}_0$ și pentru fiecare $z \in \mathbf{S}_0$ se adaugă cîte o stare suplimentară $z \oplus \theta$, din mulțimea $\mathbf{M}_0$.

8. Se adaugă la $\mathscr{R}(\mathscr{A})$ relațiile dintre stările automatului care apar prin introducerea de stări suplimentare și se reia algoritmul de la pasul 2.

În continuare vom da cîteva exemple pentru precizarea mai bună a modului de aplicare a metodei și pentru a se scoate în evidență posibilitățile acesteia.

Exemplul 1. Să se transforme automatul cu matricea de tranziție a stărilor din tabelul 11.3 într-un automat care respectă condiția tehnică de funcționare.

1. Mulțimea elementelor de trecere este $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$ dată de mulțimea relațiilor de trecere $\mathscr{R}(\mathscr{A})$ a automatului

$$\{\theta_1 = z_1 \oplus z_3,\ \theta_2 = z_1 \oplus z_4,\quad \theta_3 = z_1 \oplus z_2,\quad \theta_4 = z_2 \oplus z_4,\quad \theta_5 = z_3 \oplus z_4,\quad \theta_6 = z_2 \oplus z_3\}.$$

Mulțimea relațiilor $\mathscr{R}(\Theta)$ se obține prin eliminarea stărilor automatului din mulțimea relațiilor de trecere $\mathscr{R}(\mathscr{A})$. Se obține $\mathscr{R}(\Theta) = \{\theta_5 = \theta_1\theta_2,\ \theta_6 = \theta_1\theta_3,\ \theta_4 = \theta_2\theta_3\}$,

3. Deoarece sistemul de ecuații $\mathscr{R}(\Theta)$ este deja rezolvat, se poate lua mulțimea $\Theta_0 = \{\theta_1, \theta_2, \theta_3\}$.

4. Matricea de trecere rezultă imediat și este dată în tabelul 11.5.

Tabelul 11.5

z \ x	00	01	11	10
z_1	θ_1	θ_1	θ_2	θ_0
z_2	θ_3	$\theta_2\theta_3$	θ_0	θ_0
z_3	$\theta_1\theta_2$	θ_0	$\theta_1\theta_3$	θ_1
z_4	θ_0	θ_0	θ_0	$\theta_2\theta_3$

5. Automatul nu îndeplinește condiția tehnică de bună funcționare, întrucît nu toate trecerile între stări se fac prin stări vecine.

6. Mulțimea $\mathbf{M}_0 = \{z_2 \oplus \theta_2,\ z_3 \oplus \theta_2,\ z_3 \oplus \theta_3\}$.

7. Mulțimea $\mathbf{S}_0 = \{z_2,\ z_3\}$. Se pot adăuga stările suplimentare $z_5 = z_2 \oplus \theta_2$, $z_6 = z_3 \oplus \theta_2$, $z_7 = z_3 \oplus \theta_3$.

8. Adăugăm stările suplimentare z_5, z_6 și z_7. Noile relații adăugate la $\mathscr{R}(\mathscr{A})$ sînt $z_2 \oplus z_5 = \theta_2$, $z_3 \oplus z_6 = \theta_2$, $z_3 \oplus z_7 = \theta_3$, $z_4 \oplus z_5 = \theta_3$, $z_4 \oplus z_6 = \theta_1$, $z_2 \oplus z_7 = \theta_1$ și se șterg din $\mathscr{R}(\mathscr{A})$ relațiile $z_2 \oplus z_4 = \theta_2\theta_3$, $z_3 \oplus z_4 = \theta_1\theta_2$, $z_3 \oplus z_3 = \theta_1\theta_3$. Atunci $\mathscr{R}(\mathscr{A}) = \{z_1 \oplus z_3 = \theta_1,\ z_1 \oplus z_4 = \theta_2,\ z_1 \oplus z_2 = \theta_3,\ z_2 \oplus z_5 = \theta_2,\ z_3 \oplus z_6 = \theta_2,\ z_3 \oplus z_7 = \theta_3,\ z_4 \oplus z_5 = \theta_3,\ z_4 \oplus z_6 = \theta_1,\ z_2 \oplus z_7 = \theta_1\}$.

Revenind la pasul 2, rezultă că elementele θ_1, θ_2, θ_3 sînt liniar independente și trecerile dintre stări se fac prin stări vecine. Matricea de trecere este dată în tabelul 11.6, a), iar matricea de tranziție a stărilor în tabelul 11.6, b).

Tabelul 11.6

z \ x	00	01	11	10
z_1	θ_0	θ_1	θ_2	θ_0
z_2	θ_3	θ_2	θ_0	θ_0
z_3	θ_2	θ_0	θ_3	θ_1
z_4	θ_0	θ_0	θ_0	θ_3
z_5	—	θ_3	—	θ_2
z_6	θ_1	—	—	—
z_7	—	—	θ_1	—

a

z \ x	00	01	11	10
z_1	z_1	z_3	z_4	z_1
z_2	z_1	z_5	z_2	z_2
z_3	z_6	z_3	z_7	z_1
z_4	z_4	z_4	z_4	z_5
z_5	—	z_4	—	z_2
z_6	z_4	—	—	—
z_7	—	—	z_2	—

b

Luînd $\theta_1 = 001$, $\theta_2 = 010$, $\theta_3 = 100$ și $z_1 = 000$, se obține asignarea elementelor intermediare $z_2 = 100$, $z_3 = 001$, $z_4 = 010$, $z_5 = 110$, $z_6 = 011$, $z_7 = 101$.

Exemplul 2. Aplicînd metoda de adăugare de stări suplimentare, să se demonstreze că automatul cu matricea de tranziție a stărilor din tabelul 11.7 îndeplinește condiția tehnică de bună funcționare.

Tabelul 11.7

z \ x	00	01	11	10
z_1	z_1	z_2	z_1	z_2
z_2	z_1	z_2	z_3	z_2
z_3	z_4	z_4	z_3	z_4
z_4	z_4	z_4	z_1	z_4

Mulțimea relațiilor automatului $\mathscr{R}(\mathscr{A})$ și mulțimea elementelor de trecere Θ sînt:

$$\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}, \quad \mathscr{R}(\mathscr{A}) = \{\theta_1 = z_1 \oplus z_2,\ \theta_2 = z_2 \oplus z_3,\ \theta_3 = z_3 \oplus z_4,\ \theta_4 = z_1 \oplus z_4\}.$$

2. Mulțimea relațiilor $\mathscr{R}(\Theta) = \{\theta_1\ \theta_2\ \theta_3\ \theta_4 = z_1\ z_2\ z_3\ z_4 = \theta_0\}$, de unde se obțin $\theta_3 = \theta_1$, $\theta_4 = \theta_2$.

4. Matricea de trecere este dată în tabelul 11.8. Rezultă deci că automatul îndeplinește condiția tehnică de bună funcționare.

Tabelul 11.8

z \ x	00	01	11	10
z_1	θ_0	θ_1	θ_0	θ_1
z_2	θ_1	θ_0	θ_2	θ_0
z_3	θ_1	θ_1	θ_0	θ_1
z_4	θ_0	θ_0	θ_2	θ_0

Exemplul 3. Să se transforme automatul cu matricea de tranziție a stărilor din tabelul 11.9, a) într-un automat care respectă condiția tehnică de bună funcționare.

Tabelul 11.9

z \ x	00	01	11	10
z_1	z_1	z_2	z_4	z_3
z_2	z_3	z_2	z_2	z_2
z_3	z_3	z_4	z_2	z_3
z_4	z_1	z_4	z_4	z_2

a

z \ x	00	01	11	10
z_1	θ_0	θ_1	θ_2	θ_3
z_2	$\theta_1\theta_3$	θ_0	θ_0	θ_0
z_3	θ_0	$\theta_2\theta_3$	$\theta_1\theta_3$	θ_0
z_4	θ_2	θ_0	θ_0	$\theta_1\theta_2$

b

1. Mulțimile elementelor și relațiilor de trecere sînt

$$\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}, \quad \mathscr{R}(\mathscr{A}) = \{\theta_1 = z_1 \oplus z_2,\ \theta_2 = z_1 \oplus z_4,\ \theta_3 = z_1 \oplus z_3,\ \theta_4 = z_2 \oplus z_3,\ \theta_5 = z_3 \oplus z_4,\ \theta_6 = z_2 \oplus z_4\}.$$

2. Mulțimea relațiilor între elementele de trecere este $\mathscr{R}(\Theta) = \{\theta_4 = \theta_1\ \theta_3,\ \theta_5 = \theta_2\ \theta_3,\ \theta_6 = \theta_1\ \theta_2\}$.

3. $\Theta_0 = \{\theta_1, \theta_2, \theta_3\}$.

4. Matricea de trecere este dată în tabelul 11.9, b).

5. Automatul nu îndeplinește condiția tehnică de bună funcționare.

6. Mulțimea $\mathbf{M}_0 = \{z_2 \oplus \theta_3,\ z_3 \oplus \theta_2,\ z_2 \oplus \theta_2\}$.

7. Mulțimea $\mathbf{S}_0 = \{z_2, z_3\}$. Se pot adăuga stările suplimentare $z_5 = z_2 \oplus \theta_3$, $z_6 = z_3 \oplus \theta_2$, $z_7 = z_2 \oplus \theta_2$. Există o singură posibilitate de adăugare de stări suplimentare.

8. Relațiile $\mathscr{R}(\mathscr{A})$ devin

$$\mathscr{R}(\mathscr{A}) = \{\theta_1 = z_1 \oplus z_2,\ \theta_2 = z_1 \oplus z_4,\ \theta_3 = z_1 \oplus z_3,\ z_2 \oplus z_5 = \theta_3,\ z_2 \oplus z_5 = \theta_1;\ z_3 \oplus z_6 = \theta_2,\ z_4 \oplus z_6 = \theta_3,\ z_2 \oplus z_7 = \theta_2,\ z_4 \oplus z_7 = \theta_1\}.$$

Rezultă că automatul obținut în urma adăugării de stări suplimentare prezentat în tabelul 11.10 îndeplinește condiția tehnică de bună funcționare.

Tabelul 11.10

z \ x	00	01	11	10
z_1	z_1	z_2	z_4	z_3
z_2	z_5	z_2	z_2	z_2
z_3	z_3	z_6	z_5	z_3
z_4	z_1	z_4	z_4	z_7
z_5	z_3	—	z_2	—
z_6	—	z_4	—	—
z_7	—	—	—	z_2

Pentru asignarea elementelor intermediare luăm de exemplu: $\theta_1 = 001$, $\theta_2 = 010$, $\theta_3 = 100$. În plus, fixăm și una din stările stabile. Pentru exemplificare luăm $z_1 = 000$.

Se va obține asignarea din tabelul 11.11.

Tabelul 11.11

z \ x	00	01	11	10
000	000	001	010	100
001	101	001	001	001
100	100	110	101	100
010	000	010	010	011
101	100	—	001	—
110	—	010	—	—
011	—	—	—	001

11.4. Funcțiile de lucru ale elementelor intermediare

Etapa următoare a sintezei constă în scrierea funcțiilor de lucru ale elementelor intermediare induse de funcția de tranziție a automatului. Dacă $Y_1, Y_2, \ldots, Y_n$ sînt elementele intermediare asignate și $X_1, X_2, \ldots, X_m$ elementele de intrare și dacă notăm cu $y_1, y_2, \ldots, y_n$ și respectiv $x_1, x_2, \ldots, x_m$ variabilele asociate contactelor elementelor respective, atunci funcționarea fiecăruia dintre elementele intermediare va fi descrisă de cîte o funcție booleană de variabilele $x_1, x_2, \ldots, x_m$ și $y_1, y_2, \ldots, y_n$:

$$\bar{Y}_i = F_i(x_1, x_2, \ldots, x_m; y_1, y_2, \ldots, y_n), i = 1, 2, \ldots, n,$$

de forma

$$F_i(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n) = \sum x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_m^{\alpha_m} y_1^{\beta_1} y_2^{\beta_2} \ldots y_n^{\beta_n},$$

unde suma se extinde la toate perechile din tabelul de asignare a elementelor intermediare $(\alpha_1, \ldots, \alpha_m)$, $(\beta_1, \beta_2, \ldots, \beta_n)$ pentru care în coloana corespunzătoare elementului Y_i se găsește 1. De exemplu, dacă notăm elementele intermediare cu Y_1, Y_2 și Y_3, iar elementele de intrare cu X_1 și X_2, tabelul 11.11 se transformă în tabelul 11.12.

Tabelul 11.12

$Y_1Y_2Y_3$ \ X_1X_2	00	01	11	10
000	000	001	010	100
001	101	001	001	001
100	100	110	101	100
010	000	010	010	011
101	100	—	001	—
110	—	010	—	—
011	—	—	—	001

care conduce la următoarele funcții de lucru pentru Y_1, Y_2, Y_3:

$$\bar{Y}_1 = x_1^1x_2^0y_1^0y_2^0y_3^0 + x_1^0x_2^0y_1^0y_2^0y_3^1 + x_1^0x_2^0y_1^1y_2^0y_3^0 + x_1^0x_2^1y_1^1y_2^0y_3^0 + x_1^1x_2^1y_1^1y_2^0y_3^0 + x_1^1x_2^0y_1^1y_2^0y_3^0 + x_1^0x_2^0y_1^1y_2^0y_3^1,$$

$$\bar{Y}_2 = x_1^1x_2^1y_1^0y_2^0y_3^0 + x_1^0x_2^1y_1^1y_2^0y_3^0 + x_1^0x_2^1y_1^0y_2^1y_3^0 + x_1^1x_2^1y_1^0y_2^1y_3^0 + x_1^1x_2^0y_1^0y_2^1y_3^0 + x_1^0x_2^1y_1^1y_2^1y_3^0,$$

$$\bar{Y}_3 = x_1^0x_2^1y_1^0y_2^0y_3^0 + x_1^0x_2^0y_1^0y_2^0y_3^1 + x_1^0x_2^1y_1^0y_2^0y_3^1 + x_1^1x_2^1y_1^0y_2^0y_3^1 + x_1^1x_2^0y_1^0y_2^0y_3^1 + x_1^1x_2^1y_1^1y_2^0y_3^0 + x_1^1x_2^0y_1^0y_2^1y_3^0 + x_1^1x_2^1y_1^1y_2^0y_3^1 + x_1^0x_2^0y_1^1y_2^1y_3^1,$$

corespunzătoare următoarelor mulțimi de perechi:

$\mathcal{M}(Y_1) = \{(10,000), (00,001), (00,100), (01,100), (11,100), (10,100), (00,101)\}$,

$\mathcal{M}(Y_2)$: $\{(11,000), (01,100), (01,010), (11,010), (10,010), (01,110)\}$,

$\mathcal{M}(Y_3)$: $\{(01,000), (00,001), (01,001), (11,001), (10,001), (11,100), (10,010), (11,101), (10,011)\}$.

În același mod se stabilesc funcțiile de lucru pentru elementele de ieșire, plecînd de la matricea de ieșire, așa cum s-a arătat atunci cînd s-a descris metoda de sinteză a lui Huffman.

11.5. Eliminarea hazardului din circuitele de comutație

Prin hazard în circuitele de comutație se înțelege riscul de defecțiune în funcționare. De aici rezultă că hazardul este legat de realizarea concretă a schemei, de natura elementelor care compun circuitul respectiv. Pot exista două cauze care conduc la hazard în funcționare. Prima dintre acestea este uzura elementelor componente care conduce la degradarea parametrilor acestora peste limitele admisibile. A doua cauză, foarte frecventă, este descrierea incorectă a realității. De regulă, contactele releelor sînt considerate bipoziționale, fără să se țină seama de trecerea de la starea deschis la starea închis și invers, care este de fapt o a treia stare a contactului și dacă nu este luată în considerare poate provoca defecțiuni în funcționare.

În afara defecțiunilor provocate de funcționarea incorectă a contactelor aceluiași releu există o altă categorie de defecțiuni provocate de contacte ale unor relee diferite. Nu ne vom ocupa de acest caz, întrucît vom presupune că este respectată condiția tehnică de bună funcționare sau condițiile tehnice de bună funcționare, dacă sînt impuse mai multe astfel de condiții.

În privința hazardului introdus de contactele aceluiași releu se disting trei cazuri. Două dintre aceste cazuri corespund situației cînd în circuitul considerat există mai multe contacte ale aceluiași releu și al treilea caz situației cînd schema conține numai un singur contact al unui releu. Să analizăm mai întîi acest ultim caz. În teorie se presupune că trecerea unui contact de la o poziție la alta se face fără vibrație, ceea ce nu corespunde întotdeauna realității. Dacă după ce a trecut de la starea 0 la starea 1, sau invers, contactul revine înapoi, și dacă timpul de reacție a elementelor care compun scheme este mai mic decît acest timp de vibrare, poate rezulta o funcționare incorectă a schemei. Acest tip de hazard poartă numele de *hazard dinamic*.

Existența mai multor contacte ale aceluiași releu este o sursă serioasă de defecțiuni. Două sau mai multe contacte de același fel, de închidere sau de deschidere, conduc la risc de hazard, atunci cînd nu se închid sau nu se deschid simultan. O situație similară se poate produce în circuitele electronice cînd timpii de propagare a unui același semnal pe diferite trasee sînt diferiți. Hazardul de acest tip a fost numit *hazard esențial* de către [35] Unger care a dat o regulă de recunoaștere a acestuia pe matricea de tranziție a stărilor. După Unger există posibilitatea de hazard esențial atunci cînd trei schimbări consecutive ale unui element primar conduc automatul în altă stare decît o singură schimbare a elementului respectiv. Nu vom insista asupra acestui tip de hazard întrucît nu există un alt remediu pentru hazardul esențial decît egalizarea timpilor de propagare [35].

În fine, ultimul tip de hazard este așa-numitul *hazard static*, denumit și *hazard de continuitate*, care poate fi provocat de existența într-un circuit a două variabile complementare. Fie, de exemplu, x^1 și x^0. După cît se știe, între aceste variabile trebuie să existe relațiile $x^1 x^0 = 0$ și $x^1 + x^0 = 1$. Elementele utilizate în practică nu îndeplinesc aceste condiții ideale care presupun că sînt posibile numai combinațiile $x^1 x^0 = 01$ și 10 și că sînt imposibile combinațiile $x^1 x^0 = 00$ și 11.

În realitate, trecerea de la combinația 10 a variabilelor $x^1 x^0$ la combinația 01 se face trecînd printr-una din combinațiile 00 sau 11, adică întîi se desface contactul de lucru x^1 ajungîndu-se la combinația 00 și apoi se face contactul x^0 ajungîndu-se la combinația 01 sau mai întîi se face contactul x^0 ajungîndu-se la combinația 11 și apoi se desface contactul x^1 pentru a se ajunge la combinația 01. Pentru trecerea de la combinația 01 la combinația 10 se poate petrece ceva similar. Apariția combinațiilor tranzitorii creează riscul de defecțiune.

Releele pentru care starea tranzitorie este 11 vor fi numite relee *FD*, în timp ce releele pentru care starea tranzitorie este 00 vor fi denumite relee *DF*.

Există în principiu două metode de eliminare a hazardului în astfel de cazuri. Prima metodă constă în descrierea reală a circuitelor [24] iar a doua în efectuarea sintezei plecînd de la expresia funcției de lucru care conține toți implicanții primi.

Vom prezenta în continuare cîteva dintre metodele uzuale de eliminare a hazardului. În teoria clasică, bivalentă, se asociază contactelor de relee variabile cu două valori 0 și 1. În [24] se propune asocierea contactelor de relee variabile cu trei valori, 0 cînd contactul este în repaus, 1 cînd contactele este în poziția de lucru și 2 cînd contactul este în poziția intermediară. Într-o astfel de situație în descrierea funcționării circuitului considerat se poate ține seama de stările tranzitorii.

Să considerăm, de exemplu, funcțiile de lucru

$$f_1(a, b, x) = a^1 x^0 + b^1 x^1,$$

$$f_2(a, b, x) = (a^1 + x^1)(b^1 + x^0),$$

care din punct de vedere al funcționării ideale sînt identice, așa cum se vede în tabelul 11.13. Se poate afirma atunci că

schemele din fig. 11.1 sînt realizări ale oricăreia dintre funcțiile $f_1(a, b, x)$, $f_2(a, b, x)$, dacă avem în vedere funcționarea

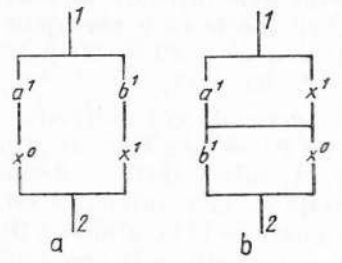

Fig. 11.1

ideală. Obținerea uneia dintre scheme sau a alteia depinde de metoda de sinteză folosită.

Tabelul 11.13

a	b	x	$f_1(a,b,x)$	$f_2(a,b,x)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Pentru a urmări funcționarea schemelor în cazul real este necesar să considerăm variabilele x^1 și x^0 ca independente și în plus să facem și ipoteze asupra tipului releului X. În tabelul 11.14 este dată funcționarea schemelor din fig. 11.1 în cazul real. În coloana 1 am trecut valorile lui $f_1(a, b, x)$ în ipoteza că releul X este de tipul DF, iar în coloana 2 valorile lui $f_1(a, b, x)$, în ipoteza că X este de tipul FD. Cu liniuță am marcat în fiecare caz situațiile care nu se pot întîlni : $x\bar{x} = 11$ pentru coloana 1 și $x\bar{x} = 00$ pentru coloana 2. La fel ca în cazul precedent în coloana 3 am trecut valorile lui $f_2(a, b, x)$ în ipoteza că releul x este de tipul DF, iar în coloana 4 valorile lui $f_2(a, b, x)$ în ipoteza că x este de tipul FD.

Tabelul 11.14

a	b	x^1	x^0	$f_1(a, b, x)$ $f_2(a, b, x)$			
				1	2	3	4
0	0	0	0	0	—	0	—
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	—	0	—	1
0	1	0	0	0	—	0	—
0	1	0	1	0	0	0	0
0	1	1	0	1	1	1	1
0	1	1	1	—	1	—	1
1	0	0	0	0	—	0	—
1	0	0	1	1	1	1	1
1	0	1	0	0	0	0	0
1	0	1	1	—	1	—	1
1	1	0	0	0	—	1	—
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	—	1	0	1

Urmărind funcționarea schemelor din fig. 11.1 în cazul real, se observă că schema 11.1, *a*) realizează numai funcția $f_1(a, b, x)$, în timp ce schema 11.1, *b*) realizează numai funcția $f_2(a, b, x)$, ceea ce reprezintă o restricție față de cazul ideal. Pe de altă parte, se observă că schema 11.1, *a*) prezintă hazard de funcționare, în ipoteza că releul X este de tipul DF (coloana 1), la combinația $ab = 11$ în starea tranzitorie $x^1x^0 = 00$ de trecere de la $x^1x^0 = 01$ la $x^1x^0 = 10$ și invers. Similar, schema 11.1, *b*) prezintă hazard de funcționare, în ipoteza că releul x este de tipul FD (coloana 4), la combinația $ab = 00$ în starea tranzitorie $x^1x^0 = 11$ de trecere de la $x^1x^0 = 01$ la $x^1x^0 = 10$ și invers.

Din acest exemplu rezultă următoarele concluzii :

1. Hazardul static este caracteristic schemelor reale și naturii elementelor care o compun. Pentru aceeași schemă se poate elimina hazardul prin simpla schimbare a unor elemente de un tip cu elemente similare de tip diferit.

2. Hazardul static se poate depista prin punerea în evidență a stărilor de tranziție, deci prin luarea în considerare a unei a treia stări pentru elementele bipoziționale și studierea lor ca elemente tripoziționale.

Acest procedeu ne furnizează o altă metodă de eliminare a hazardului. Vom numi această metodă de eliminarea hazardului *metoda descrierii reale* [21].

În continuare vom exemplifica eliminarea hazardului din cazul precedent, aplicînd metoda descrierii reale. Pentru simplitate vom lua în considerare funcționarea reală numai pentru releul X. În realitate situații similare pot exista și pentru A și B.

Vom nota cu 1 starea de lucru, cu 0 starea de repaus și cu 2 starea tranzitorie. Pentru eliminarea hazardului este suficient să impunem condițiile $f(1, 1, 2) = 1$ și $f(0, 0, 2) = 0$ pentru oricare dintre funcțiile f_1 și f_2. Din tabelul 11.14, ținînd seama că combinația $x^1x^0 = 10$ corespunde stării 1 a contactului x, $x^1x^0 = 01$ corespunde stării 0 a lui x, și $x^1x^0 = 00$ sau $x^1x^0 = 11$ corespund, după caz, stării 2 a lui x, se obține

$$f(a, b, x) = a^0b^1x^1 + a^1x^0 + a^1b^1x^0 + a^1b^1x^1 + a^1b^1x^2$$

și ținînd seama că $x^0 + x^1 + x^2 = 1$, se obține expresia redusă

$$f(a, b, x) = a^1x^0 + b^1x^1 + a^1b^1,$$

care conduce la schema din fig. 11.2.

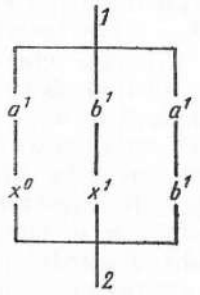

Fig. 11.2

Se obține astfel o nouă metodă de eliminare a hazardului pe care o vom numi *metoda implicanților primi redondanți*. Într-adevăr, prin adăugarea la funcția $f_1(a, b, x) = a^1x^0 + b^1x^1$ a implicantului prim a^1b^1, care în cazul funcționării ideale este redondant, se obține expresia $f(a, b, x) = a^1x^0 + b^1x^1 + a^1b^1$ care conduce la schema 11.2, fără hazard de continuitate.

Putem deci enunța următoarea proprietate :

3. Hazardul static se elimină, indiferent de tipul elementelor constituente, dacă funcția de lucru a circuitului se scrie ca reuniune a tuturor implicanților săi primi.

Nu vom da demonstrația acestei afirmații aici, ne mulțiumim cu exemplificarea făcută, proprietatea fiind cunoscută.

CAPITOLUL 12

SINTEZA MULTIPOLILOR FĂRĂ REPETIȚIE

12.1. Generalități

După cum am văzut în capitolele precedente, funcțiile de tranziție și de ieșire conduc la sisteme de funcții de lucru pentru elementele intermediare și elementele de ieșire. Pentru realizarea concretă a automatelor este deci necesar să se pună la punct metode de sinteză a circuitelor secvențiale corespunzătoare acestor sisteme de funcții de lucru.

Ideea cea mai simplă este să se facă sinteza funcție cu funcție, constituind dipolii corespunzători printr-una din metodele cunoscute. Acest procedeu este neeficient din punct de vedere economic, întrucît funcțiile respective conțin, de regulă, aceleași variabile și s-ar mări peste măsură numărul de contacte ale aceluiași releu, ceea ce ar conduce, în afară de unele dificultăți tehnice de realizabilitate, la scumpirea instalațiilor și la creșterea riscului de defecțiuni. Este mai natural deci să se dezvolte metode de sinteză a unor multipoli, cu un număr minim de elemente care să reprezinte sistemul de funcții de lucru și nu funcții individuale. În realitate, după cum se va arăta, se poate întotdeauna asocia unui sistem de funcții booleene o singură funcție, echivalentă din punctul de vedere al sintezei cu sistemul de funcții asociat.

O primă caracteristică a metodei de sinteză ce va fi prezentată în acest capitol constă în faptul că va trata problema sintezei multipolilor și nu numai a circuitelor

secvențiale bipolare [27, 28, 32]. O a doua caracteristică a metodei propuse este caracterul de optimalitate. Criteriul de optimizare folosit va fi numărul minim de contacte. În acest sens vom numi schema care realizează o funcție de lucru sau un sistem de funcții de lucru acea sau acele scheme care realizează funcția sau sistemul de funcții de lucru cu număr minim de contacte. În fine, o a treia caracteristică a metodei este generalitatea. Prin această metodă nu se urmărește realizarea unor scheme de un anume tip : scheme serie — paralel sau scheme în punte. Metoda conduce la cea mai simplă schemă, în sensul arătat, indiferent dacă schema respectivă este de un tip sau de altul.

În principiu, metoda cuprinde următorii pași : simplificarea funcțiilor de lucru, recunoașterea funcțiilor care se realizează prin scheme fără repetiție, sinteza multipolilor dacă ne găsim în cazul precedent, asocierea unei funcții extinse dacă nu sîntem în cazul precedent pentru a se ajunge la o funcție care se realizează prin scheme fără repetiție, efectuarea sintezei și revenirea la variabilele inițiale.

Metoda de sinteză propusă se bazează pe proprietățile unor coduri asociate funcțiilor de lucru în forma lor simplificată. O proprietate remarcabilă a funcțiilor care se realizează prin schema fără repetiție este că le corespund coduri grupale. Proprietăți ale acestor coduri conduc la un criteriu de recunoaștere a funcțiilor care se realizează prin scheme fără repetiție.

12.2. Funcția de lucru a unui multipol

Vom presupune ca funcțiile de lucru care fac obiectul sintezei sînt simplificate și luăm reuniunea tuturor implicanților lor primi ca expresie de bază. Ne vom ocupa mai întîi de multipolii fără repetiție.

Definiția 1. Spunem că o funcție booleană F se realizează prin *scheme fără repetiție* dacă există o schemă dipolară $\mathscr{S}_F$ în care fiecare variabilă, cuprinsă, în F, apare o

singură dată și dacă F este funcția de lucru a dipolului $\mathscr{S}_F$. În plus, $\bar{x}$ dacă x este una din variabilele lui F, atunci în $\mathscr{S}_F$ nu apare $\bar{x}$ în același timp cu x.

Fie $F = \bigcup_i A_i$, $i \in \mathbf{I}$, funcția F, care se realizează prin scheme fără repetiție, unde A_i este un produs de variabile

$$A_i = a_{i_1}^{\alpha_{i_1}} a_{i_2}^{\alpha_{i_2}} \dots a_{i_k}^{\alpha_{i_k}},$$

iar $\mathbf{I}$ o mulțime de indici.

Definiția 2. A_i este un *implicant prim* al funcției F, dacă

$$A_i = 1 \Rightarrow F = 1$$

și dacă această implicație încetează în cazul cînd eliminăm din A_i cel puțin o variabilă.

Definiția 3. Se numește *formă disjunctivă normală redusă* sau, simplu, *formă redusă a funcției booleene* F, reuniunea tuturor implicanților primi ai lui F.

Fie $\mathscr{S}_F$ schema fără repetiție care realizează funcția F. Vom conveni să introducem în $\mathscr{S}_F$ încă o latură a care unește polii 1 și 2, pe care o vom numi pe scurt *latura sursei* (fig. 12.1) și vom considera multipolul $\mathscr{M}_F$ format din

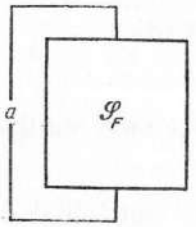

Fig. 12.1

schema $\mathscr{S}_F$ în serie cu a. Vom înmulți apoi fiecare implicant prim A_i al lui F cu a. Fie

$$F_a = \bigcup_i a A_i, \quad i \in \mathbf{I}.$$

Dacă notăm $aA_i = B_i$, atunci F_a se poate scrie sub forma

$$F_a = \bigcup_i B_i, \quad i \in \mathbf{I}.$$

Ținînd seama de faptul că fiecare A_i reprezintă un drum în $\mathscr{S}_F$ care unește nodurile 1 și 2 și că mulțimea implicanților primi $\{A_i\}$, $i \in \mathbf{I}$, reprezintă mulțimea tuturor drumurilor care unesc pe 1 cu 2 în $\mathscr{S}_F$, atunci $B_i = aA_i$ va reprezenta un ciclu simplu în schema din fig. 12.1, iar mulțimea $\{B_i\}$, $i \in \mathbf{I}$, va reprezenta mulțimea tuturor ciclurilor simple ale schemei din fig. 12.1 care conțin latura a.

Fie $\{B_i\}$, $i \in \mathbf{I}$, mulțimea implicanților primi din F_a la care adăugăm implicantul $\emptyset$, care nu conține nici o variabilă. În mulțimea $\{B_i\}$ introducem operația $\oplus$ definită astfel:

1) $B_i \oplus B_i = \emptyset$, $(\forall) i \in \mathbf{I}$.

2) Dacă $B_i = bB_i'$ și $B_j = bB_j'$, unde b este produsul variabilelor comune lui B_i și B_j iar B_i' și B_j' nu conțin variabile comune, atunci $B_i \oplus B_j = B_i' \oplus B_j'$.

Plecînd de la mulțimea $\{B_i\}$, $i \in \mathbf{I}$, vom construi o mulțime $\mathbf{B}$ de cicluri simple, după următoarea regulă:

1) $B_i \in \mathbf{B}$, $(\forall) i \in \mathbf{I}$.

2) Dacă $B_k = B_i \oplus B_j$, $i, j \in \mathbf{I}$, și dacă nu există $B_l \in \mathbf{B}$ astfel încît $B_k + B_l = B_l$, atunci B_k va fi inclus în $\mathbf{B}$ și în plus orice $B_l \in \mathbf{B}$ pentru care $B_l + B_k = B_k$ va fi eliminat din $\mathbf{B}$.

Exemplu. Pentru funcția

$$F = ac + bd + acd + bec$$

după înmulțirea cu variabila p se obține

$$Fp = pac + pbd + pacd + pbec.$$

Mulțimea $\mathbf{B}$ va fi următoarea:

$$\mathbf{B} = \{pac, pbd, pacd, pbec, abcd, ced, abe, \emptyset\},$$

unde

$$abcd = pac \oplus pbd, \quad ced = pac \oplus pacd, \quad abe = pac \oplus pbec,$$

existînd și alte moduri de obținere a elementelor respective.

Schema $\mathscr{S}_F$ asociată funcției F, completată cu latura p, este dată în fig. 12.2.

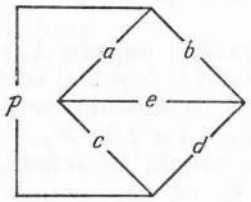

Fig. 12.2

Vom demonstra acum:

Teorema 1. *Mulțimea* **B** *formează grup comutativ cu operația* $\oplus$, *grupul ciclurilor simple ale multipolului* $\mathscr{M}_F$.

Într-adevăr:

$(B_i \in \mathbf{B},\ B_j \in \mathbf{B}) \Rightarrow B_i \oplus B_j \in \mathbf{B}$,

$B_i \oplus B_j = B_j \oplus B_i$,

$B_i \oplus B_i = \emptyset \Rightarrow (B_i)^{-1} = B_i \in \mathbf{B}$,

$B_i \oplus X = B_j \Rightarrow X = B_i \oplus B_j \in \mathbf{B}$.

Grupul **B** are $l - n + 1$ generatori, unde l este numărul de laturi din $\mathscr{M}$ și n numărul de noduri pentru multipoli simplu conecși și $l - n + s$ generatori pentru scheme multiplu conexe [27].

Să considerăm funcția booleană F_i formată din suma ciclurilor simple din **B** care conțin variabila i și fie F funcția obținută prin înlăturarea lui i din F_i.

Teorema 2. *Dacă* i *este una din laturile unui multipol oarecare,* $\mathscr{M}$, *iar* F_i *este suma ciclurilor simple care conțin pe* i, *atunci funcția* F *obținută prin înlăturarea variabilei* i *din fiecare factor al lui* F_i, *este funcția de lucru a dipolului* $\mathscr{D}_i$ *obținut din* $\mathscr{M}$ *prin fixarea drept poli a nodurilor la care este conectat* i.

Demonstrație. Fiecare ciclu simplu, care conține latura i, conține un implicant prim al dipolului $\mathscr{D}_i$, deoarece reprezintă, în afara laturii i, un drum continuu între nodurile la care este conectat i, în care fiecare latură apare odată și numai odată. Pe de altă parte, după cum s-a arătat orice implicant prim completat cu latura i devine un ciclu simplu al multipolului $\mathscr{M}$.

Din teorema 2 rezultă că grupul **B** caracterizează nu numai dipolul $\mathscr{S}_F$, a cărui funcție de lucru este F, ci multipolul $\mathscr{M}$ însuși, întrucît funcția de lucru a oricărui dipol din $\mathscr{M}$ poate fi obținută ca sumă de elemente din **B**.

Definiție. Dacă $B_i \in \mathbf{B}$, $i \in \mathbf{I}$, atunci funcția booleană $F = \sum_{i \in \mathbf{I}} B_i$ va fi numită *funcție de lucru a multipolului* $\mathscr{M}$, corespunzător mulțimii **B**.

Ținînd seama de teorema 2 și de proprietățile funcțiilor booleene, rezultă următoarea teoremă.

Teorema 3. *Funcția de lucru* F *a unui multipol* $\mathscr{M}$, *fără repetiție se scrie în funcție de funcțiile de lucru ale dipolilor* $\mathscr{D}_i \in \mathscr{M}$ *sub forma*

$$F = \sum_{i=1}^{l} a_i F_i,$$

unde l *este numărul de laturi în* $\mathscr{M}$, a_i *laturile multipolului, iar* F_i *funcțiile de lucru ale dipolilor* $\mathscr{D}_i$, *corespunzători laturilor* a_i, $i = 1, 2, \ldots, l$.

12.3. Un algoritm de sinteză a dipolilor fără repetiție

Fie $F = \Phi(a_1, a_2, \ldots, a_l)$ o funcție de lucru a unui dipol fără repetiție. Prin sinteza dipolului înțelegem construcția acestuia plecînd de la funcția de lucru dată.

Vom nota $F^* = \overline{\Phi}\,(\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_l)$ duala funcției F. Funcției F^* i se asociază [28] un grup $\mathbf{B}^*$ al implicanților primi ai funcției F^*, dual cu grupul **B** al implicanților primi ai funcției F. Grupul $\mathbf{B}^*$ are $n-1$ generatori pentru dipoli simplu conecși.

Dacă A_k^*, $k \in \{1, 2, \ldots, n\}$, sînt implicanţii primi ai lui F^* şi dacă B_k este dualul lui A_k^*, $B_k = (A_k^*)^*$, atunci există următoarele implicaţii :

1° $F = 1 \Rightarrow B_k = 1$.

2° $B_k = 0 \Rightarrow F = 0$.

3° A_k^* încetează de a mai fi implicant al funcţiei F^* dacă se înlătură din el cel puţin o variabilă.

Implicaţiile precedente rezultă din egalitatea $F = \prod_{k=1}^{m} B_k$ şi din proprietatea lui A_k^* de a fi implicant prim al funcţiei F^*.

Să considerăm funcţia F a unui multipol $\mathcal{M}$:

$$F = \Phi(a_1, a_2, \ldots, a_l) = \sum_{i=1}^{l} a_i F_{a_i}.$$

Definiţie. Vom numi *duala funcţiei* F a multipolului $\mathcal{M}$ şi o vom nota cu F^* funcţia

$$F^* = \sum_{i=1}^{l} a_i F_{a_i}^*,$$

unde $F_{a_i}^*$ este duala funcţiei de lucru F_{a_i} a dipolului obţinut din $\mathcal{M}$ prin eliminarea laturii a_i, care are poli nodurile la care este conectată a_i. Ţinînd seama de relaţiile 1°, 2°, 3°, rezultă că variabilele din orice implicant prim A^* al unei funcţii de lucru $F_{a_i}^*$ constituie o secţiune în funcţia de lucru dual㠄 F_{a_i}. Deci printre implicanţii A^* ai funcţiei $F_{a_i}^*$ se găsesc şi nodurile dipolului $\mathcal{S}_{F_{a_i}}$ corespunzător.

Această proprietate ne permite să dăm un algoritm de sinteză a dipolilor (multipolilor) fără repetiţie.

Descrierea algoritmului. 1°. Fiind dată funcţia de lucru F a unui dipol fără repetiţie, aceasta se scrie mai întîi sub forma disjunctivă normală redusă.

2° Se calculează duala F^* a funcţiei F care se scrie de asemenea sub forma normală disjunctivă redusă.

3° Se înmulţeşte fiecare implicant prim din forma redusă cu latura p, obţinîndu-se ciclurile simple ale multipolului asociat $\mathcal{M}_F$.

4° Se alege numărul maxim de cicluri simple liniar independente. Se va obţine o bază formată din $n - 1$ cicluri, care este o bază în grupul $\mathbf{B}^*$ asociat lui F^*.

5° Se compun ciclurile simple ale bazei obţinute două cîte două, după legea $\oplus$ în scopul reducerii numărului total de variabile din bază. După fiecare operaţie se înlocuieşte unul din ciclurile simple care intervin în relaţie cu rezultatul compunerii, dacă acesta este mai scurt. Se realizează astfel o schimbare a bazei de generatori a grupului $\mathbf{B}^*$.

6° Se continuă procesul de compunere pînă cînd fiecare variabilă apare în cel mult două cicluri. O astfel de bază de generatori a grupului $\mathbf{B}^*$ constituie $n - 1$ noduri ale multipolului $\mathcal{M}_F$, al n-lea nod fiind format de variabilele care apar cîte o singură dată şi se obţine prin compunerea tuturor implicanţilor primi din bază.

Exemplu. Fie :

$F = ag + bh + ci + adh + bdg + cfh + bfi + adfi + cfdg,$

$F^* = abc + ghi + abfi + adhi + ghfc + gdbc + adhfc + gdbfi,$

$F_p^* = pabc + pghi + pabfi + padhi + pghfc + pgdbc + padhfc + pgdbfi.$

Un sistem liniar independent de generatori ai lui $\mathbf{B}^*$ este format din mulţimea de implicanţi primi $\{pabc, pghi, padhi, pghfc\}$. Compunînd pe *pghi* cu *padhi* şi cu *pghfc*, se obţine o nouă bază cu un număr mai mic de variabile :

$$pabc,\ pghi,\ adg,\ cfi.$$

în care fiecare variabilă apare cel mult de două ori şi deci reprezintă patru noduri ale multipolului $\mathcal{M}_F$. Al cincilea nod este dat de expresia

$$pabc \oplus pghi \oplus adg \oplus cfi = bhdf.$$

Deci $\mathcal{M}_F$ are nodurile date de mulţimea de implicanţi primi

$$pabc,\ pghi,\ adg,\ cfi,\ bhdf,$$

care conduc la multipolul din fig. 12.3, în care după eliminarea laturii p se obţine dipolul care realizează funcţia F.

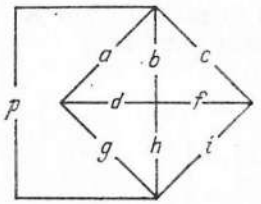

Fig. 12.3

12.4. Codurile grupale generate de funcțiile booleene

Vom asocia funcțiilor booleene coduri binare care vor juca un rol important în sinteza multipolilor. Aceste coduri vor fi introduse pe o cale diferită de cea privind simplificarea, ținînd seama de scopul nou pe care-l urmărim aici.

Ca de obicei, funcțiile booleene studiate se scriu în prealabil sub forma normală disjunctivă redusă, deci ca reuniune a tuturor implicanților lor primi. În plus, dacă $F = \bigcup_i A_i$ este funcția studiată și dacă în F apare x alături de $\bar{x}$, în codurile asociate, acestea vor fi considerate diferite adică lui x și $\bar{x}$ le vor corespunde coloane distincte în codul $\mathbf{K}_F$ asociate funcției F, ceea ce diferențiază esențial acest cod de codurile asociate anterior.

Fie deci $F = \bigcup_i A_i$ forma redusă a funcției F. Presupunem mai întîi că funcția F se realizează prin scheme fără repetiție și fie $\mathbf{B}$ codul ciclurilor simple ale multipolului $\mathbf{M}_F$ care realizează funcția F. Dacă $B_i = b_{i1}\ b_{j2} \ldots b_{im}$ este unul dintre ciclurile simple ale multipolului M_F, convenim să-i asociem un vector binar $\alpha_i = (\alpha_{i1}\ \alpha_{i2} \ldots \alpha_{il})$ după următoarea regulă :

1° Coloanele vectorului α_i corespund variabilelor distincte din $\mathbf{B}$. Există deci atîtea coloane în α_i, cîte variabile diferite apar în $\mathbf{B}$, cu convenția ca x și $\bar{x}$ să fie considerate distincte, dacă apar împreună în F.

2° $a_k \in B_i \Rightarrow \alpha_{ik} = 1$,

$a_k \notin B_i \Rightarrow \alpha_{ik} = 0$.

Definiția 1. Vom numi *cod binar* $\mathbf{K}_F$ generat de funcția F matricea cu l coloane și 2^m linii :

$$\mathbf{K}_F = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_q \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1l} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2l} \\ \vdots & \vdots & & \vdots \\ \alpha_{q1} & \alpha_{q2} & \ldots & \alpha_{ql} \end{pmatrix},$$

construită cu ajutorul vectorilor α_i, unde $q = 2^m$, m fiind numărul de linii liniar independente din $\mathbf{K}$.

Între liniile α_i, α_j ale codului $\mathbf{K}_F$ introducem operația $\oplus$ definită astfel :

$$\alpha_i \oplus \alpha_j = (\alpha_{i1} \oplus \alpha_{j1},\ \alpha_{i2} \oplus \alpha_{j2} \ldots, \alpha_{il} \oplus \alpha_{jl}),$$

unde $\alpha_{ik} \oplus \alpha_{jk}$ este suma modulo 2 a elementelor α_{ik} și α_{jk}.

Se vede imediat că dacă $\alpha_m = \alpha_i \oplus \alpha_j$ și dacă $\alpha_i \Leftrightarrow B_i$, $\alpha_j \Leftrightarrow B_j$, atunci $\alpha_m \Leftrightarrow B_m$, unde $B_m = B_i \oplus B_j$.

Proprietate. *Liniile codului* $\mathbf{K}_F$ *pot fi ordonate astfel încît* $\alpha_{2^s+t} = \alpha_{2^s} \oplus \alpha_t$ *pentru orice* $0 \leqslant s \leqslant m-1$, *unde* $m = l - n + 1$ *este numărul de linii liniar independente în* $\mathbf{K}_F$ *și corespund la* m *implicanți primi liniar independenți.*

Dar cum $\mathbf{B}$ este grup cu operația $\oplus$, atunci și mulțimea liniilor codului $\mathbf{K}_F$ va forma grup cu operația $\oplus$, izomorf cu grupul $\mathbf{B}$.

Definiția 2. Codul binar $\mathbf{K}_{F^*}$ generat de funcția F^*, cu l coloane și 2^{n-1} linii va fi numit *codul dual al codului* $\mathbf{K}_F$.

Liniile codului $\mathbf{K}_{F^*}$ formează grup cu operația $\oplus$, izomorf cu grupul $\mathbf{B}^*$ asociat funcției F^*.

Exemplu. Funcției $F = ac + bd + aed + bec$, căreia îi corespunde schema din fig. 12.2, i se asociază codul $\mathbf{K}_F$ din tabelul următor :

$$\mathbf{K}_F = \begin{array}{c} \begin{matrix} a & b & c & d & e & p \end{matrix} \\ \left(\begin{matrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{matrix} \right) \end{array},$$

în care se respectă proprietatea precedentă. Această proprietate ne dă o regulă de construcție a codului $\mathbf{K}_F$.

12.5. Proprietăți ale codurilor grupale generate de funcțiile care se realizează prin scheme fără repetiție

Vom introduce următoarele transformări de echivalență :

T_1 — permutarea liniilor,

T_2 — permutarea coloanelor,

T_3 — adunarea modulo 2 a tuturor liniilor cu o linie dată din $\mathbf{K}_F$.

Definiția 3. Două coduri grupale **K** și **K*** se numesc *echivalente*, dacă matricea unuia poate fi obținută din matricea celuilalt printr-un număr finit de transformări de echivalență.

Definiția 4. Doi multipoli fără repetiție se numesc *echivalenți*, dacă se poate găsi o corespondență biunivocă între mulțimile dipolilor conținute în cei doi multipoli, astfel încît funcțiile de lucru pentru dipolii din oricare pereche, care se găsesc în corespondența astfel stabilită, să fie identice.

Teorema 1. *Dacă unui multipol $\mathcal{M}$ conex, fără repetiție, îi corespunde codul grupal* **K**, *atunci oricărui cod* **K'** *echivalent cu* **K** *îi corespunde același multipol $\mathcal{M}$ sau un multipol echivalent cu el.*

Demonstrație. Fie α_i o linie arbitrară a codului **K** și $B_i = b_{i1} b_{i2} \dots b_{im}$ ciclul din $\mathcal{M}$ corespunzător. Să arătăm că transformările de echivalență nu modifică pe B_i.

T_1 permută liniile codului **K**. În urma unei transformări de tipul T_1, α_i se regăsește în alt loc al lui **K**, deci B_i rămîne în **B**.

T_2 permută coloanele lui **K**, deci permută variabilele b_{ik} între ele, ceea ce nu modifică pe B_i.

T_3 este transformarea $\alpha_i \oplus \alpha$, α fiind o linie dată, iar α_i o linie oarecare a lui **K**; ținînd seama că liniile lui **K** formează grup cu operația $\oplus$, ea realizează un automorfism al lui **K**, deci reproduce, într-o altă ordine, toate liniile lui **K**. Atunci și B_i se va regăsi în **B**.

În continuare vom introduce următoarele transformări de structură :

T_4 — eliminarea uneia sau mai multor coloane,

T_5 — eliminarea tuturor liniilor care conțin 1 într-o coloană dată.

Pentru a vedea cum operează transformările de structură vom demonstra cîteva teoreme.

Teorema 2. *Dacă $\mathcal{M}$ este un multiplu fără repetiție și* **K** *codul grupal corespunzător lui $\mathcal{M}$, atunci prin aplicarea unei transformări de tipul T_4 se obține un cod* **K'**, *căruia i se asociază multipolul $\mathcal{M}'$, obținut din $\mathcal{M}$ prin scurtcircuitarea laturii corespunzătoare coloanei eliminate.*

Demonstrație. Fie $\mathcal{D}$ un dipol al lui $\mathcal{M}$ și F funcția de lucru corespunzătoare lui $\mathcal{D}$. Fie b variabila care corespunde coloanei eliminate din **K**. Efectul acestei transformări asupra ciclurilor B_i este înlăturarea lui b din fiecare ciclu B_i care îl conține. Dar acest lucru este echivalent cu înlocuirea lui b în F cu 1, ceea ce este același lucru cu scurtcircuitarea lui b în $\mathcal{D}$ și respectiv în $\mathcal{M}$, q.e.d.

Teorema 3. *Dacă $\mathcal{M}$ este un multipol fără repetiție și* **K** *codul grupal corespunzător, atunci prin aplicarea unei transformări de tipul T_5 se obține un cod* **K'** *căruia îi corespunde multipolul $\mathcal{M}'$, obținut din $\mathcal{M}$ prin eliminarea laturii corespunzătoare coloanei care a făcut obiectul transformării.*

Demonstrație. Fie $\mathcal{D}$ un dipol al lui $\mathcal{M}$ și F funcția de lucru corespunzătoare lui $\mathcal{D}$. Aplicăm o transformare de tipul T_5 codului **K**. Fie b variabila corespunzătoare coloanei care determină transformarea. Efectul asupra mulțimii **B** este înlăturarea din aceasta a tuturor ciclurilor B_i care conțin variabila b și lăsarea restului de cicluri B_i nemodificate. Dar acest lucru este echivalent cu egalarea cu **0** a lui b în F, ceea ce este același lucru cu înlăturarea lui b din $\mathcal{D}$ și respectiv din $\mathcal{M}$.

Evident orice transformare de structură asupra multipolului $\mathcal{M}$ fără repetiție care conduce la multipoli fără repetiție este echivalentă cu aplicarea unui șir finit de transformări de tipul 4° și 5° asupra codului **K** asociat.

12.6. Un criteriu de recunoaștere a funcțiilor care se realizează prin scheme fără repetiție

Folosind proprietățile codurilor grupale, vom demonstra un criteriu de recunoaștere a funcțiilor booleene care se realizează prin schema fără repetiție și, pe această bază, vom da un nou algoritm pentru sinteza schemelor fără repetiție.

Dacă F și F^* sînt două funcții booleene duale, vom nota cu $\mathbf{B}$ și $\mathbf{B}^*$ grupurile generate de implicanții primi ai lui F și respectiv F^* și prin $\mathbf{K}$ și $\mathbf{K}^*$ codurile generate de funcțiile respective.

Fie $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{il})$, $i \in \mathbf{I}$, liniile matricei $\mathbf{K}$ și $\beta_k = (\beta_{k1}, \beta_{k2}, \ldots, \beta_{kl})$, $k \in \mathbf{J}$, liniile matricei $\mathbf{K}^*$, unde l este numărul de variabile, n numărul de noduri, $\mathbf{I} = \{1, 2, \ldots, 2^m\}$, $\mathbf{J} = \{1, 2, \ldots, 2^{n-1}\}$, iar $m = l - n + 1$.

Fie $\beta_i\beta_k = (\beta_{i1}\beta_{k1}, \beta_{i2}\beta_{k2}, \ldots, \beta_{il}\beta_{kl})$ produsul logic al liniilor β_i și β_k din $\mathbf{K}^*$. Vom nota $\gamma_s = \beta_i\beta_k$, unde s parcurge o mulțime de indici $\mathbf{S}$ și fie $\mathbf{K}_2^*$ matricea formată de vectorii γ_s, $s \in \mathbf{S}$. Fie de asemenea

$$\beta_i \oplus \beta_k = (\beta_{i1} \oplus \beta_{k1}, \quad \beta_{i2} \oplus \beta_{k2}, \ldots, \beta_{il} \oplus \beta_{kl})$$

și

$$\bigoplus_{j=1}^{l} \alpha_{ij}\beta_{kj} = \alpha_{i1}\beta_{k1} \oplus \alpha_{i2}\beta_{k2} \oplus \ldots \oplus \alpha_{il}\beta_{kl}.$$

Teorema 1. *O condiție necesară pentru ca funcția F să se realizeze prin scheme fără repetiție este dată de relația*

$$\bigoplus_{j=1}^{l} \alpha_{ij}\beta_{kj} = 0, \qquad i \in \mathbf{I}, \quad k \in \mathbf{J},$$

denumită relația de ortogonalitate între linii.

Demonstrație. Este suficient să demonstrăm teorema pentru o bază a codului $\mathbf{K}^*$. Vom considera o linie a codului $\mathbf{K}^*$ corespunzătoare unui implicant prim al lui F^*. Fie β_k această linie și fie $l_1, l_2, \ldots, l_s$ coloanele în care linia β_k are 1. Atunci există egalitățile

$$\beta_{kj} = 1, \qquad j \in \{l_1, l_2, \ldots, l_s\},$$

$$\beta_{kj} = 0, \qquad j \notin \{l_1, l_2, \ldots, l_s\}.$$

Evident, latura sursei se găsește printre coloanele din mulțimea $\{l_1, l_2, \ldots, l_s\}$, fie chiar l_s, fără să restringem generalitatea. Produsul $A = b_{l_1} b_{l_2} \ldots b_{l_{s-1}}$, unde b_{l_j} este variabila corespunzătoare coloanei l_j din $\mathbf{K}^*$, este implicant prim al lui F^*. Dar atunci rezultă implicațiile

$$A = 1 \Rightarrow b_{l_1} = b_{l_2} = \ldots = b_{l_{s-1}} = 1 \Rightarrow F^* = 1,$$

$$A^* = 0 \Rightarrow b_{l_1} = b_{l_2} = \ldots = b_{l_{s-1}} = 0 \Rightarrow F = 0,$$

scrise condensat astfel: $A \Rightarrow F^*$ și respectiv $\bar{A}^* \Rightarrow \bar{F}$.

Din implicația $\bar{A}^* \Rightarrow \bar{F}$ rezultă că fiecare implicant prim al funcției F conține cel puțin o variabilă b_r, $r \in \{l_1, \ldots, l_{s-1}\}$, deci fiecare linie α_i a codului $\mathbf{K}$, corespunzătoare unui implicant prim al lui F, conține 1 măcar într-una din coloanele $l_1, l_2, \ldots, l_{s-1}$, corespunzătoare variabilelor $b_{l_1}, b_{l_2}, \ldots, b_{l_{s-1}}$, și de asemenea conține 1 în coloana l_s, corespunzătoare laturii sursei b_{l_s}. Să arătăm că numărul de 1 în coloanele $l_1, l_2, \ldots, l_{s-1}$ este impar. În caz contrar există cel puțin o linie a codului $\mathbf{K}$, fie α_i, care are 1 în coloana l_s și un număr par de 1 în coloanele $l_1, l_2, \ldots, l_{s-1}$.

Fie $p_1, p_2, \ldots, p_{2r}$, $p_j \in \{l_1, l_2, \ldots, l_{s-1}\}$, $j = 1, 2, \ldots, 2r$, coloanele în care α_i are 1. Atunci

$$q \in \{p_1, p_2, \ldots, p_{2r}\} \Rightarrow \alpha_{iq} = 1.$$

Deoarece A este un implicant prim al lui F^*, înlăturarea oricărei variabile b_q, $q \in \{l_1, l_2, \ldots, l_{s-1}\}$ din A conduce la anularea implicației $A \Rightarrow F^*$, ceea ce arată că există, pentru fiecare q, un implicant prim A_q al lui F care conține pe b_q și nu mai conține nici o altă variabilă din A. Dacă această proprietate nu ar fi satisfăcută, b_q ar putea fi eliminat din A păstrîndu-se implicația $A \Rightarrow F^*$, adică A nu ar fi implicant prim.

Fie α_q linia din codul $\mathbf{K}$ corespunzătoare implicantului A_q al lui F, $q \in \{p_1, p_2, \ldots, p_{2r}\}$ cu proprietatea de mai sus. Dar atunci există implicațiile

$$\begin{aligned} &j = q \Rightarrow \alpha_{qj} = 1, \\ &j \neq q \Rightarrow \alpha_{qj} = 0, \end{aligned} \qquad j \in \{p_1, p_2, \ldots, p_{2r}\},$$

Rezultă că linia α_m din $\mathbf{K}$ obținută din liniile α_{p_1}, $\alpha_{p_2}, \ldots, \alpha_{p_{2r}}$ și α_i prin relația $\alpha_m = \alpha_{p_1} \oplus \alpha_{p_2} \oplus \ldots$

$\ldots \oplus \alpha_{p_{2r}} \oplus \alpha_i = (\alpha_{m1}, \alpha_{m2}, \ldots, \alpha_{ml})$ se bucură de proprietatea :

$$\alpha_{mj} = 0 \quad \text{pentru} \quad j = q,\ q \in \{p_1, p_2, \ldots, p_{2r}\},$$

$$\alpha_{mj} = 1 \quad \text{pentru} \quad j = l_s \quad \text{(latura sursei)}.$$

Dar din felul în care a fost ales A_q, $q \in \{l_1, l_2, \ldots, l_{s-1}\}$, rezultă că $\alpha_{mj} = 0$ pentru orice alt $j \in \{l_1, l_2, \ldots, l_{s-1}\}$, adică există un implicant A_m al lui F care nu conține nici o variabilă $b \in \{b_{l_1}, b_{l_2}, \ldots, b_{l_{s-1}}\}$ și deci nu se verifică implicația $A^* = 0 \Rightarrow F = 0$, ceea ce ar conduce la concluzia că A nu este implicant prim pentru F^*, contrar presupunerii. Deci teorema se verifică pentru implicantul prim A al lui F^*. Întrucît implicantul prim A a fost ales arbitrar, rezultă că teorema se verifică pentru orice implicant prim al lui F^*.

Fie $\{\beta_1, \beta_2, \ldots, \beta_r\}$ o mulțime de linii din $\mathbf{K}^*$, corespunzătoare unor implicanți primi ai lui F^* :

$$\beta_k = (\beta_{k1}, \beta_{k2}, \ldots, \beta_{kl}), \qquad k = 1, 2, \ldots, r;$$

conform demonstrației, există relația de ortogonalitate

$$\sum_{j=1}^{l} \alpha_{ij}\beta_{kj} = 0, \qquad i \in \mathbf{I}, \quad k \in \{1, 2, \ldots, r\},$$

unde $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{il})$ este o linie oarecare a codului $\mathbf{K}$, iar $\mathbf{I}$ mulțimea de indici ai codului $\mathbf{K}$. Să arătăm că relația de ortogonalitate se verifică pentru orice linie a codului $\mathbf{K}^*$. Fie

$$\nu = \sum_{k=1}^{r} \lambda_k \beta_k$$

o combinație liniară a liniilor $\beta_1, \beta_2, \ldots, \beta_r$, $\nu = (\nu_1, \ldots \nu_l)$. Vom obține succesiv :

$$\alpha_i \nu = \sum_{k=1}^{r} \lambda_k \alpha_i \beta_k = \sum_{k=1}^{r} \lambda_k (\alpha_{i1}\beta_{k1}, \alpha_{i2}\beta_{k2}, \ldots, \alpha_{il}\beta_{kl})$$

și deci

$$\sum_{j=1}^{l} \alpha_{ij}\nu_j = \sum_{k=1}^{r} \lambda_k \sum_{j=1}^{l} \alpha_{ij}\beta_{kj} = 0,$$

adică relația de ortogonalitate este adevărată pentru orice linie a codului $\mathbf{K}^*$.

Teorema 2. *Fie l linii $\gamma_1, \gamma_2, \ldots, \gamma_l$ din $\mathbf{K}_2^*$ cu proprietatea :*

$$\gamma_{pq} = 1 \quad \textit{pentru} \quad q = p;\ p, q = 1, 2, \ldots, l,$$

$$\gamma_{pq} = 0 \quad \textit{pentru} \quad q \neq p,$$

l fiind numărul de coloane în $\mathbf{K}_2^*$. Se poate găsi o mulțime $\{\beta_1, \beta_2, \ldots, \beta_m\}$ de linii în $\mathbf{K}^*$ astfel încît fiecare $\gamma \in \{\gamma_1, \gamma_2, \ldots, \gamma_l\}$ să fie produsul logic a două linii $\beta_i, \beta_j \in \{\beta_1, \beta_2, \ldots, \beta_m\}$ și în plus pentru fiecare $s \in \{1,2,\ldots, l\}$ există două și numai două linii $\beta_i, \beta_j \in \{\beta_1, \beta_2, \ldots, \beta_m\}$ care au 1 în coloana s. Atunci liniile $\beta_1, \beta_2, \ldots, \beta_m$ sînt liniar dependente și oricare $m-1$ dintre acestea sînt liniar independente.

Demonstrație. După cum am mai arătat, elementele codului $\mathbf{K}_2^*$ sînt definite astfel :

$$\gamma_s = \beta_i\beta_j \Rightarrow \gamma_s \in \mathbf{K}_2^*, \qquad \beta_i, \beta_j \in \mathbf{K}^*,$$

ceea ce definește pe $\mathbf{K}_2^*$ ca produs al codului $\mathbf{K}^*$ cu el însuși :

$$\mathbf{K}_2^* = \mathbf{K}^*\,\mathbf{K}^*.$$

Deoarece printre implicanții primi ai funcției F^* se regăsesc nodurile dipolului $\mathscr{S}_F$ care realizează funcția F, pentru fiecare variabilă a din $\mathscr{S}_F$ există doi implicanți primi A și B ai lui F^* cu proprietatea

$$A = A'\,a, \qquad B = B'\,a$$

și astfel încît A' și B' nu mai au variabile comune.

Observație. Am presupus că am lăsat la o parte laturile conectate în paralel eliminînd coloanele identice din codul $\mathbf{K}^*$.

Implicanților primi A și B ai lui F^* le corespund în $\mathbf{K}^*$ linii cu 1 în coloana corespunzătoare lui a și componente diferite în restul de coloane. Altfel spus, dacă β_i este linia din $\mathbf{K}^*$ corespunzătoare lui A și β_k linia corespunzătoare lui B, atunci produsul logic $\beta_i\beta_k$ are componenta

corespunzătoare lui a egală cu 1 și restul de componente nule. Cu aceasta am demonstrat că există $\gamma_1, \gamma_2, \ldots, \gamma_l$ în $\mathbf{K}_2^*$ și $\beta_1, \beta_2, \ldots, \beta_m$ în $\mathbf{K}^*$ cu proprietățile din enunț.

Vom demonstra acum partea a doua a teoremei.

Dacă $i_1, i_2, \ldots, i_r$ sînt coloanele în care β_i are 1, atunci β_i se poate scrie

$$\beta_i = \gamma_{i1} \oplus \gamma_{i2} \oplus \ldots \oplus \gamma_{ir}, \qquad i = 1, 2, \ldots, m,\ r \leqslant l,$$

și deoarece pentru fiecare $s \in \{1, 2, \ldots, l\}$ există două și numai două linii din mulțimea $\{\beta_1, \beta_2, \ldots, \beta_l\}$ care au 1 în coloana s, atunci în mulțimea $\{\beta_i, i \in [1, m]\}$ fiecare γ_s apare de două și numai de două ori, de unde rezultă relația de dependență liniară

$$\beta_1 \oplus \beta_2 \oplus \ldots \oplus \beta_m = 0$$

între liniile $\beta_1, \beta_2, \ldots, \beta_m$. Să observăm că dacă $\beta_1, \beta_2, \ldots, \beta_{m-1}$ nu ar fi liniar independente, s-ar obține

$$\lambda_1\beta_1 \oplus \lambda_2\beta_2 \oplus \ldots \oplus \lambda_{m-1}\beta_{m-1} = 0,$$

unde nu toți λ_i, $i \in \{1, 2, \ldots, m-1\}$ sînt nuli. Aceasta conduce la o relație de dependență liniară între $\gamma_1, \gamma_2, \ldots, \gamma_l$:

$$\lambda_1\gamma_1 \oplus \lambda_2\gamma_2 \oplus \ldots \oplus \lambda_l\gamma_l = 0$$

în care nu toți λ_i sînt nuli, întrucît prin lipsa lui β_m din combinația liniară a elementelor β există cel puțin un element $\gamma \in \{\gamma_1, \ldots, \gamma_l\}$ care apare o singură dată în mulțimea $\{\beta_i, i \in \{1, 2, \ldots, m-1\}$, ceea ce este absurd, $\gamma_1, \gamma_2, \ldots, \gamma_l$ fiind liniar independente.

Să considerăm $m-1$ linii $\beta_1, \beta_2, \ldots, \beta_{m-1}$ ale codului $\mathbf{K}^*$, liniar independente, obținute în condițiile teoremei 2. Vom putea demonstra următoarea teoremă.

Teorema 3. *Orice linie β a codului* $\mathbf{K}^*$ *este o combinație liniară a liniilor $\beta_1, \beta_2, \ldots, \beta_{m-1}$, obținute în teorema 2.*

Demonstrație. Fie $\beta_1, \beta_2, \ldots, \beta_{m-1}$ cele $m-1$ linii ale codului $\mathbf{K}^*$, liniar independente, $m-1$ fiind numărul maxim care se bucură de această proprietate. Fiecare

$\beta_k \in \{\beta_1, \beta_2, \ldots, \beta_{m-1}\}$ poate fi scris ca o combinație liniară a mulțimii de linii $\gamma_1, \ldots, \gamma_l$ ale codului $\mathbf{K}_2^*$, în condițiile teoremei 2 :

$$\beta_k = \alpha_1\gamma_1 \oplus \alpha_2\gamma_2 \oplus \ldots \oplus \alpha_l\gamma_l,$$

unde α_i iau valori egale cu 1 sau 0. Fie $\alpha_p = 1$. Atunci putem scrie

$$\gamma_p = \beta_k \oplus \alpha_1\gamma_1 \oplus \ldots \oplus \alpha_{p-1}\gamma_{p-1} \oplus \alpha_{p+1}\gamma_{p+1} \oplus \ldots \oplus \alpha_l\gamma_l.$$

Putem înlocui deci baza $\{\gamma_1, \gamma_2, \ldots, \gamma_l\}$ prin baza $\{\beta_k, \gamma_1, \ldots, \gamma_{p-1}, \gamma_{p+1}, \ldots, \gamma_l\}$ și mai departe prin inducție putem arăta că în baza $\{\gamma_1, \gamma_2, \ldots, \gamma_l\}$ se pot înlocui $m-1$ dintre aceste linii cu liniile $\beta_1, \beta_2, \ldots, \beta_{m-1}$, obținînd o nouă bază $\{\beta_1, \beta_2, \ldots, \beta_{m-1}, \gamma_m, \gamma_{m+1}, \ldots, \gamma_l\}$. Dar deoarece orice linie β a codului $\mathbf{K}^*$ poate fi exprimată ca o combinație liniară de baza $\{\gamma_1, \gamma_2, \ldots, \gamma_l\}$, ea va putea fi exprimată și ca o combinație liniară a bazei noi, echivalente cu aceasta. Fie deci $\beta \in \mathbf{K}^*$:

$$\beta = \lambda_1\beta_1 \oplus \lambda_2\beta_2 \oplus \ldots \oplus \lambda_{m-1}\beta_{m-1} \oplus \lambda_m\gamma_m \oplus \ldots \oplus \lambda_l\gamma_l.$$

Pe de altă parte, dacă $\alpha_i = (\alpha_{i1}\alpha_{i2} \ldots \alpha_{il})$ este o linie a codului $\mathbf{K}$, atunci, ținînd seama de teorema 1,

$$\Xi\ \alpha_{ij}\beta_{kj} = 0, \qquad (\forall)\ k = 1, 2, \ldots, m-1.$$

Din aceasta și din faptul că γ_s are numai componenta s egală cu 1, restul fiind egale cu zero, se obțin relațiile reduse

$$\alpha_{i_m}\lambda_m \oplus \alpha_{i_{m+1}}\lambda_{m+1} \oplus \ldots \oplus \alpha_{i_l}\lambda_l = 0$$

pentru orice linie α_i a codului $\mathbf{K}$.

Fie $j_1, j_2, \ldots, j_r$, $r \leqslant l - m + 1$, coloanele pentru care $\lambda_{j1} = \lambda_{j2} = \ldots = \lambda_{jr} = 1$, restul elementelor fiind nule. Relația precedentă devine

$$\alpha_{ij_1} \oplus \alpha_{ij_2} \oplus \ldots \oplus \alpha_{ij_r} = 0$$

pentru orice linie α_i din $\mathbf{K}$, de unde rezultă că pentru orice linie α_i există un număr par de componente $\alpha_{ij} = 1$, $j \in \{j_1, j_2, \ldots, j_r\}$. Vom numi această relație relație de dependență liniară.

Evident $r \leqslant 1 - m + 1 \neq 1$, altfel $\alpha_{il} = 0$ pentru orice i, contrar presupunerii. Să arătăm că dacă relația de mai sus nu se verifică pentru $l - m + 1 = p$, ea nu se verifică nici pentru $l - m + 1 = p + 1$. Într-adevăr, să presupunem că nu există coduri cu $l - m + 1 = p$ pentru care acest lucru se întîmplă, dar există coduri cu $l - m + 1 = p + 1$ pentru care această proprietate are loc. Orice astfel de cod se reduce la coduri cu $l - m + 1 = p$ prin eliminarea unei coloane oarecare. Să considerăm toate liniile codului **K** în care există 1 într-o coloană arbitrară j_k. Pentru ca relația de dependență liniară să se verifice este suficient să se verifice pentru aceste linii, întrucît acestea generează codul. Presupunem deci că relația de dependență liniară se verifică pentru o linie a codului **K** în care $\alpha_{ij_k}=1$; atunci rezultă că $\alpha_{ij_1} \oplus \ldots \oplus \alpha_{ij_{k-1}} \oplus \alpha_{ij_{k+1}} \oplus \ldots \oplus \alpha_{ij_r} = 1$. Dar pentru fiecare linie a codului **K**, în care $\alpha_{ij_k} = 1$, există o linie α_q cu proprietatea

$$\alpha_{ij_s} = \alpha_{qj_s} \quad \text{pentru} \quad s \neq k,$$

$$\alpha_{ij_k} = 1, \quad \alpha_{qj_k} = 0.$$

După înlocuire se obține

$$\alpha_{qj_1} \oplus \ldots \oplus \alpha_{qj_{k-1}} \oplus \alpha_{qj_{k+1}} \oplus \ldots \oplus \alpha_{qj_r} = 1$$

și deoarece $\alpha_{qj_k} = 0$, se obține mai departe

$$\alpha_{qj_1} \oplus \ldots \oplus \alpha_{qj_{k-1}} \oplus \alpha_{qj_k} \oplus \alpha_{qj_{k+1}} \oplus \ldots \oplus \alpha_{qj_r} = 1,$$

contrar relației de dependență liniară. Dar atunci relațiile reduse pot fi adevărate pentru orice s numai dacă $\lambda_s = 0$, $s = m, \; m + 1, \ldots, l$, ceea ce demonstrează lema și în plus $l - m + 1 = 0$, adică $l = m - 1$. Se obține deci proprietatea că numărul maxim de linii liniar independente în codul **K*** este egal cu numărul de variabile ale funcției F.

Acum sîntem în măsură să demonstrăm o teoremă care ne dă o condiție necesară și suficientă pentru ca o funcție booleeană să se realizeze prin scheme fără repetiție și care ne conduce la un nou algoritm de sinteză în acest caz.

Teorema 4. *Condiția necesară și suficientă pentru ca o funcție booleană F să se realizeze prin scheme fără repetiție este dată de următoarele două puncte :*

1) *Între liniile $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{il})$ ale codului* **K** *generat de funcția F și liniile $\beta_k = (\beta_{k1}, \beta_{k2}, \ldots, \beta_{kl})$ ale codului* **K*** *generat de funcția F^* există relația de ortogonalitate*

$$\underset{j}{\Xi}\, \alpha_{ij}\beta_{kj} = 0, \qquad (\forall)\ \alpha_i \in \mathbf{K}, \qquad \beta_k \in \mathbf{K}^*.$$

2) *Există l linii $\gamma_1, \gamma_2, \ldots, \gamma_l$ în codul $\mathbf{K}_2^* = \mathbf{K}^*\mathbf{K}^*$ și m linii $\beta_1, \beta_2, \ldots, \beta_m$ în* **K*** *astfel încît :*

$$\begin{aligned} &\gamma_{pq} = 1 \quad \text{pentru} \quad p = q, \\ &\gamma_{pq} = 0 \quad \text{pentru} \quad p \neq q, \end{aligned} \qquad p, q = 1, 2, \ldots, l.$$

Pentru fiecare $\gamma_s \in \{\gamma_1, \gamma_2, \ldots, \gamma_l\}$ există două linii $\beta_i, \beta_j \in \{\beta_1, \beta_2, \ldots, \beta_m\}$ și numai două cu proprietatea $\gamma_s = \beta_i\beta_j$. Altfel spus, fiecare variabilă este cuprinsă în două și numai două dintre nodurile $\beta_1, \beta_2, \ldots, \beta_m$.

Demonstrație. Necesitatea rezultă din teorema 1 ținînd seama că relația de ortogonalitate este chiar condiția de necesitate impusă de această teoremă. Suficiența rezultă din teorema 2, în care se demonstrează că pentru funcțiile care se realizează prin scheme fără repetiție există γ_1, $\gamma_2, \ldots, \gamma_l$ și respectiv $\beta_1, \beta_2, \ldots, \beta_m$ care verifică pe 2) din teorema de față. Dar în condițiile 2) se poate construi o schemă $\mathbf{S}_F$ care are noduri chiar pe $\beta_1, \beta_2, \ldots, \beta_m$ și în care nodul β_i și nodul β_j se interconectează printr-o latură dacă produsul $\beta_i\beta_j$ este diferit de vectorul nul. Dacă $\beta_i\beta_j = \gamma_s$, atunci latura care leagă nodul β_i cu β_j se notează cu a_s, unde s este coloana γ_s diferită de zero.

Mai rămîne să arătăm că schema obținută este chiar schema căutată. Din fiecare nod β_i vor pleca atîtea laturi cîte elemente egale cu 1 sînt în linia β_i. Într-adevăr, dacă $\beta_{is} = 1$, atunci conform lui 2) există încă un β_k și numai unul pentru care $\beta_{ks} = 1$. Cum s a fost ales arbitrar, rezultă că pentru fiecare $\beta_{is} = 1$, $s \in \{1, 2, \ldots, l\}$, există o latură în $\mathscr{S}_F$. Adică nodul β_i corespunde unui implicant prim în F^* și, conform algoritmului din 12.3, $\mathscr{S}_F$ este chiar schema care realizează funcția F.

Să considerăm acum cazul general în care în codul $\mathbf{K}^*$ pot apărea și coloane identice.

Teorema 5. *Mulțimea schemelor fără repetiție care realizează funcția booleană F este dată de implicanții primi ai funcției*

$$\Phi(\beta_1, \beta_2, \ldots, \beta_r) = \prod_{s=1}^{l} \sum_{j \in \mathbf{I}_s} \gamma_s^j$$

denumită funcția asociată schemei, unde $\mathbf{J}_s$ este mulțimea tuturor indicilor j pentru care γ_s^j are 1 în coloana s a codului $\mathbf{K}_2^*$ și 0 în restul de coloane.

Demonstrație. Fiecare implicant prim al funcției Φ conține exact l linii γ_s^j, cîte una pentru $s = 1, 2, \ldots, l$. Înlocuind liniile γ_s^j cu produsele corespunzătoare $\beta_i\beta_k$, se obțin nodurile schemei căutate.

Sîntem în măsura să dăm algoritmul de recunoaștere a funcțiilor booleene care se realizează prin scheme fără repetiție și algoritmul de sinteză corespunzător.

Descrierea algoritmului. 1. Dacă F este funcția booleană dată, se construiește funcția duală F^*.

2. Se adaugă latura sursei pe care o notăm cu p și se construiește grupul $\mathbf{B}^*$ asociat funcției F^* și codul $\mathbf{K}^*$ corespunzător.

Observație. Cînd se lucrează automat, este avantajos să se construiască direct codul $\mathbf{K}^*$. La lucrul manual este avantajos să se construiască mai întîi grupul $\mathbf{B}^*$.

3. Se construiește codul $\mathbf{K}_2^* = \mathbf{K}^*\mathbf{K}^*$.

4. Se scrie funcția $\Phi(\beta_1, \beta_2, \ldots, \beta_r)$ conform teoremei 5.

5. Dacă se verifică teorema 4, fiecare implicant conduce la o schemă fără repetiție care realizează funcția F. Dacă nu, funcția F nu se realizează prin scheme fără repetiție.

Exemplu. Să se afle dacă funcția

$$F = ac + bd + axd + bxc$$

se realizează prin scheme fără repetiție și să se facă sinteza dacă este cazul.

1. $F = ab + cd + axd + bxc$.
2. $F^* = abp + cdp + axdp + bxcp + acx + bdx + abcd$.

Notînd $\beta_1 = abp$, $\beta_2 = cdp$, $\beta_3 = axdp$, $\beta_4 = bxcp$, $\beta_5 = acx$, $\beta_6 = bdx$, $\beta_7 = abcd$, codul $\mathbf{K}^*$ va fi dat de tabelul 12.1.

Observație. Nu am înscris linia nulă întrucît nu prezintă interes pentru sinteză.

3. Codul $\mathbf{K}_2^*$ este dat de tabelul 12.2.

4. $\Phi(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7) = \gamma_1\gamma_4\gamma_5\gamma_9\gamma_{10}\gamma_{19} = \beta_1\beta_2\beta_5\beta_6$.

Tabelul 12.1

	a	b	c	d	x	p
β_1	1	1	0	0	0	1
β_2	0	0	1	1	0	1
β_3	1	0	0	1	1	1
β_4	0	1	1	0	1	1
β_5	1	0	1	0	1	0
β_6	0	1	0	1	1	0
β_7	1	1	1	1	0	0

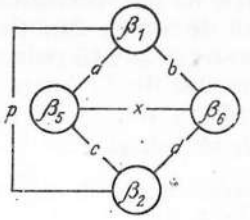

Fig. 12.4

Tabelul 12.2

	a	b	c	d	x	p
$\gamma_1\beta_1\beta_2$	0	0	0	0	0	1
$\gamma_2\beta_1\beta_3$	1	0	0	0	0	1
$\gamma_3\beta_1\beta_4$	0	1	0	0	0	1
$\gamma_4\beta_1\beta_5$	1	0	0	0	0	0
$\gamma_5\beta_1\beta_6$	0	1	0	0	0	0
$\gamma_6\beta_1\beta_7$	1	1	0	0	0	0
$\gamma_7\beta_2\beta_3$	0	0	0	1	0	1
$\gamma_8\beta_2\beta_4$	0	0	1	0	0	1
$\gamma_9\beta_2\beta_5$	0	0	1	0	0	0
$\gamma_{10}\beta_2\beta_6$	0	0	0	1	0	0
$\gamma_{11}\beta_2\beta_7$	0	0	1	1	0	0
$\gamma_{12}\beta_3\beta_4$						
$\gamma_{13}\beta_3\beta_5$						
$\gamma_{14}\beta_3\beta_6$	0	0	0	1	1	0
$\gamma_{15}\beta_3\beta_7$	1	0	0	1	0	0
$\gamma_{16}\beta_4\beta_5$	0	0	1	0	1	0
$\gamma_{17}\beta_4\beta_6$	0	1	0	0	1	0
$\gamma_{18}\beta_4\beta_7$	0	1	1	0	0	0
$\gamma_{19}\beta_5\beta_6$	0	0	0	0	1	0
$\gamma_{20}\beta_5\beta_7$	1	0	1	0	0	0
$\gamma_{21}\beta_6\beta_7$	0	1	0	1	0	0

5. Condițiile teoremei 4 se verifică, întrucît fiecare variabilă apare în două și numai două dintre liniile β ale singurului implicant prim al funcției F^*, $\beta_1\beta_2\beta_5\beta_6$. Luînd nodurile β_1, β_2, β_5, β_6, se obține schema din fig. 12.4. Nodurile sînt β_1 și β_2, deoarece între acestea este conectată latura sursei p.

CAPITOLUL 13

SINTEZA MULTIPOLILOR CU REPETIȚIE

13.1. Proprietăți ale codurilor generate de funcțiile booleene care se realizează prin scheme cu repetiție

Înainte de a trece la studiul funcțiilor booleene care se realizează prin scheme cu repetiție să mai analizăm unele proprietăți ale codurilor generate de aceste funcții.

Fie $\mathscr{F}$ o funcție booleană care se realizează prin scheme fără repetiție și **K** codul grupal generat de $\mathscr{F}$. Asupra codului **K** vom face următoarea transformare :

T_6 — *adunarea modulo 2 a două coloane.*

Exemplu. Să se adune modulo 2 coloanele a și d din codul **K** dat în tabelul 13.1 care corespunde funcției booleene

$$\mathscr{F} = ac + db + axd + bxc$$

și schemei fără repetiție din fig. 12.4.

Tabelul 13.1

	a	d	b	c	x	p	
$a\ c\ p$	1	0	0	1	0	1	
$d\ b\ p$	0	1	1	0	0	1	
$a\ d\ b\ c$	1	1	1	1	0	0	
$a\ d\ x\ p$	1	1	0	0	1	1	$= \mathbf{K}$
$d\ c\ x$	0	1	0	1	1	0	
$a\ b\ x$	1	0	1	0	1	0	
$b\ c\ x\ p$	0	0	1	1	1	1	

Se va obține codul **K'** din tabelul 13.2, în care coloana α este rezultatul compunerii modulo 2 a coloanelor a și d, $\alpha = a \oplus d$, iar funcția $\mathscr{F}$ se va reduce la funcția

$$\mathscr{F}' = \alpha c + \alpha b + x,$$

care se realizează prin schema fără repetiție din fig. 13.1.

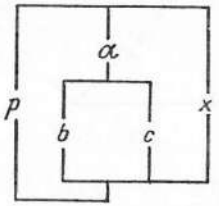

Fig. 13.1

Sînt interesante următoarele cazuri particulare :

a) Adunarea modulo 2 a coloanelor corespunzătoare la două laturi conectate în serie. Acest caz este echivalent cu scurtcircuitarea celor două laturi.

Tabelul 13.2

	α	b	c	x	p	
$\alpha\ c\ p$	1	0	1	0	1	
$\alpha\ b\ p$	1	1	0	0	1	
$b\ c$	0	1	1	0	0	
$x\ p$	0	0	0	1	1	$= \mathbf{K}'$
$\alpha\ c\ x$	1	0	1	1	0	
$\alpha\ b\ x$	1	1	0	1	0	
$b\ c\ x\ p$	0	1	1	1	1	

b) Adunarea modulo 2 a coloanelor corespunzătoare la două laturi conectate în paralele. Acest al doilea caz corespunde contopirii celor două laturi.

În cazurile analizate adunarea modulo 2 a două coloane a transformat scheme fără repetiție în scheme fără repetiție. Este interesant de demonstrat dacă această proprietate este generală sau nu.

Să trecem acum la studiul schemelor cu repetiție. Vom considera schema din fig. 13.2, obținută din schema din fig. 12.4 prin înlocuirea lui a și d prin α, adică prin

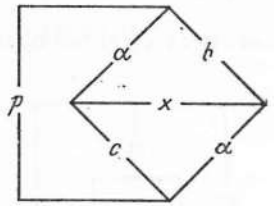

Fig. 13.2

repetarea unei variabile. Funcția de lucru a dipolului corespunzător este următoarea :

$$\mathcal{F}'' = \alpha c + \alpha b + \alpha x + bcx.$$

Codul grupal generat de funcția $\mathcal{F}''$ este dat în tabelul 13.3. Întrucît printre liniile codului **K**'' se regăsesc și liniile

Tabelul 13.3

	α	b	c	x	p	
$\alpha\ c\ p$	1	0	1	0	1	
$\alpha\ b\ p$	1	1	0	0	1	
$b\ c$	0	1	1	0	0	
$x\ p$	0	0	0	1	1	
$\alpha\ c\ x$	1	0	1	1	0	
$\alpha\ b\ x$	1	1	0	1	0	
$b\ c\ x\ p$	0	1	1	1	1	$= \mathbf{K}''$
$c\ x$	0	0	1	1	0	
$b\ x$	0	1	0	1	0	
$\alpha\ b\ c\ x\ p$	1	1	1	1	1	
α	1	0	0	0	0	
$c\ p$	0	0	1	0	1	
$b\ p$	0	1	0	0	1	
$\alpha\ b\ c$	1	1	1	0	0	

codului **K**' din tabelul 13.2, se ordonează liniile codului **K**'' astfel încît în jumătatea superioară a lui **K**'' să se regăsească liniile codului **K**'. În acest fel am pus în evidență legătura dintre cele două coduri și în același timp legătura

dintre suma modulo 2 a două coloane și repetarea laturilor respective.

Din analiza codului **K**'' rezultă că nu toate liniile pentru care $p = 1$ corespund unor implicanți ai funcției $\mathcal{F}''$. Într-adevăr, liniile 4, 12 și 13 corespund monoamelor x, c și respectiv b care nu sînt implicanți ai lui $\mathcal{F}''$. Mai mult, dacă încercăm să aplicăm teorema 2 din 12.2 pentru obținerea funcției de lucru a dipolului pentru care p este latura sursei, se obține

$$\mathcal{F}''' = x + b + c \neq \mathcal{F}''.$$

Se obține imediat un rezultat important pentru studiul funcțiilor realizabile prin scheme cu repētiție : corespondența dintre funcțiile realizabile prin scheme cu repetiție și codurile grupale generate de acestea nu este întotdeauna biunivocă.

O consecință a acestui rezultat este

Teorema 1. *O condiție suficientă pentru ca o funcție booleană $\mathcal{F}$ să se realizeze prin scheme cu repetiție este ca corespondența dintre funcția $\mathcal{F}$ și codul grupal $\mathcal{G}_F$, generat de $\mathcal{F}$, să nu fie biunivocă sau, altfel spus, în codul grupal $\mathcal{G}_F$ generat de $\mathcal{F}$, să existe linii care au 1 în coloana corespunzătoare laturii sursei și nu corespund unor implicanți ai funcției $\mathcal{F}$.*

Din păcate, condiția din teorema 1 este numai suficientă pentru realizabilitate prin scheme cu repetiție, nu și necesară, după cum rezultă din exemplul următor.

Exemplu. Se dă funcția booleană

$$F = ayc + axd + byd + bxc.$$

Să se arate că în codul grupal generat de funcția F liniile care conțin 1 în coloana corespunzătoare laturii sursei corespund unor implicanți primi ai funcției F.

Codul grupal generat de funcția F este dat în tabelul 13.4. Se verifică imediat că liniile 1, 2, 4 și 7 care conțin 1 în coloana p sînt tocmai implicanții primi ai funcției F'.

Să arătăm că funcția F din exemplul precedent nu îndeplinește criteriul de realizabilitate prin scheme fără repetiție dat de teorema 4 din 12.6. Să construim în acest

Tabelul 13.4

	a	b	c	d	x	y	p
$acyp$	1	0	1	0	0	1	1
$adxp$	1	0	0	1	1	0	1
$cdxy$	0	0	1	1	1	1	0
$cdyp$	0	1	0	1	0	1	1
$abcd$	1	1	1	1	0	0	0
$abxy$	1	1	0	0	1	1	0
$bcxp$	0	1	1	0	1	0	1

$= \mathbf{K}$

scop codul grupal **K*** asociat funcției F^*. Funcția F^* este dată de expresia redusă $F^* = ab + cd + xy + axd + ayc + byd + bxc$, iar codul **K*** asociat de tabelul 13.5.

Tabelul 13.5

		a	b	c	d	x	y	p
1	abp	1	1	0	1	0	0	1
2	cdp	0	0	1	1	0	0	1
3	$abcd$	1	1	1	0	0	0	0
4	xyp	0	0	0	0	1	1	1
5	$abxy$	1	1	0	0	1	1	0
6	$cdxy$	0	0	1	1	1	1	0
7	$abcdxyp$	1	1	1	1	1	1	1
8	$adxp$	1	0	0	1	1	0	1
9	bdx	0	1	0	1	1	0	0
10	acx	1	0	1	0	1	0	0
11	$bcxp$	0	1	1	0	1	0	1
12	adx	1	0	0	1	0	1	0
13	$bdyp$	0	1	0	1	0	1	1
14	$acyp$	1	0	1	0	0	1	1
15	bcy	0	1	1	0	0	1	0

$= \mathbf{K^*}$

Ca și în codul **K**, în codul **K*** liniilor care conțin 1 în coloana p le corespund numai implicanți ai funcției F. Întrucît codul $\mathbf{K}_2^*$ este cu un număr prea mare de linii, vom transcrie în tabelul 13.6 numai liniile care conțin un singur 1, restul de elemente fiind egale cu 0. Unele linii

ale codului $\mathbf{K}_2^*$ vor fi repetate pentru a pune în evidență liniile codului **K*** din care provin. În stînga fiecărei linii a codului $\mathbf{K}_2^*$ vor fi scrise liniile codului **K***, a căror com-

Tabelul 13.6

	a	b	c	d	x	y	p
1 2	0	0	0	0	0	0	1
1 4	0	0	0	0	0	0	1
1 9	0	1	0	0	0	0	0
1 10	1	0	0	0	0	0	0
1 12	1	0	0	0	0	0	0
1 15	0	1	0	0	0	0	0
2 4	0	0	0	0	0	0	1
2 9	0	0	0	1	0	0	0
2 10	0	0	1	0	0	0	0
2 12	0	0	0	1	0	0	0
2 15	0	0	1	0	0	0	0
4 9	0	0	0	0	1	0	0
4 10	0	0	0	0	1	0	0
4 12	0	0	0	0	0	1	0
4 15	0	0	0	0	0	1	0
9 10	0	0	0	0	1	0	0
9 12	0	0	0	1	0	0	0
9 15	0	1	0	0	0	0	0
10 12	1	0	0	0	0	0	0
10 15	0	0	1	0	0	0	0
12 15	0	0	0	0	0	1	0

$= \mathbf{K}_2^*$

punere dau linia respectivă. Calculînd funcția $\Phi(\beta_1, \beta_2, \beta_4, \beta_9, \beta_{10}, \beta_{12}, \beta_{15})$, se obține :

$$\Phi(\beta_1, \beta_2, \beta_4, \beta_9, \beta_{10}, \beta_{12}, \beta_{15}) =$$
$$= (\beta_1\beta_{10} + \beta_1\beta_{12} + \beta_{10}\beta_{12})(\beta_1\beta_9 + \beta_1\beta_{15} +$$
$$+ \beta_9\beta_{15})(\beta_2\beta_{10} + \beta_2\beta_{15} + \beta_{10}\beta_{15})(\beta_2\beta_9 + \beta_2\beta_{12})(\beta_4\beta_9 +$$
$$+ \beta_4\beta_{10} + \beta_9\beta_{10})(\beta_4\beta_{12} + \beta_4\beta_{15} + \beta_{12}\beta_{15})(\beta_1\beta_2 +$$
$$+ \beta_1\beta_4 + \beta_2\beta_4) = \beta_1\beta_2\beta_9\beta_{10}\beta_{12}\beta_{15} + \beta_1\beta_4\beta_9\beta_{10}\beta_{12}\beta_{15} +$$
$$+ \beta_2\beta_4\beta_9\beta_{10}\beta_{12}\beta_{15} + \beta_1\beta_2\beta_4\beta_9\beta_{10}\beta_{12} + \beta_1\beta_2\beta_4\beta_9\beta_{10}\beta_{15} +$$
$$+ \beta_1\beta_2\beta_4\beta_9\beta_{12}\beta_{15} + \beta_1\beta_2\beta_4\beta_{10}\beta_{12}\beta_{15}.$$

Se poate verifica relativ ușor valabilitatea relației între liniile codurilor **K** și **K***, adică se verifică prima parte a teoremei 4 din 12.6.

Nu același lucru se poate afirma despre al doilea punct al teoremei respective. Într-adevăr, conform teoremei 5 din 12.6, implicanții primi ai funcției $\Phi(\beta_1, \ldots, \beta_r)$ ar trebui să îndeplinească prevederile punctului 2 din teorema 4. Or, în exemplul de față, oricare implicant prim am lua în considerare, acesta nu verifică partea a doua a teoremei 4. De exemplu, pentru $\beta_1\beta_2\beta_9\beta_{10}\beta_{12}\beta_{15}$, unde $\beta_1 \leftrightarrow abp$, $\beta_2 \leftrightarrow cdp$ $\beta_9 \leftrightarrow bdx$, $\beta_{10} \leftrightarrow acx$, $\beta_{12} \leftrightarrow ady$, $\beta_{15} \leftrightarrow bcy$, fiecare variabilă, în afară de p, apare în trei noduri, contrar punctului 2 din teorema 4.

T_7 — *dedublarea modulo 2 a unei coloane.* Vom demonstra printr-un exemplu cum transformarea T_7 poate să conducă de la funcții care se realizează prin scheme cu repetiție la funcții care se realizează prin scheme fără repetiție.

Exemplu. Fiind dată funcția booleană

$$F = ab + ac + ad + bcd,$$

să se verifice dacă se realizează prin scheme fără repetiție. Dacă nu, să se găsească o schemă fără repetiție din care provine.

Vom construi codul grupal **K** asociat funcției booleene F, în tabelul 13.7. Întrucît liniile β_{11} β_{13} și β_{14} nu corespund unor implicanți ai funcției F,

Tabelul 13.7

		a	b	c	d	p	
1	abp	1	1	0	0	1	
2	acp	1	0	1	0	1	
3	bc	0	1	1	0	0	
4	adp	1	0	0	1	1	
5	bd	0	1	0	1	0	
6	cd	0	0	1	1	0	
7	$adcdp$	1	1	1	1	1	
8	$bcdp$	0	1	1	1	1	= **K**
9	acd	1	0	1	1	0	
10	abd	1	1	0	1	0	
11	dp	0	0	0	1	1	
12	abc	1	1	1	0	0	
13	cp	0	0	1	0	1	
14	bp	0	1	0	0	1	
15	a	1	0	0	0	0	

conform teoremei 1 funcția se realizează prin scheme cu repetiție. Printr-o permutare a liniilor codului **K**, acesta poate fi scris sub forma din tabelul 13.8.

Tabelul 13.8

	a	b	c	d	p	
abp	1	1	0	0	1	
acp	1	0	1	0	1	
bc	0	1	1	0	0	
$bcdp$	0	1	1	1	1	
acd	1	0	1	1	0	
abd	1	1	0	1	0	
dp	0	0	0	1	1	
adp	1	0	0	1	1	= **K**
bd	0	1	0	1	0	
cd	0	0	1	1	0	
$abcdp$	1	1	1	1	1	
abc	1	1	1	0	0	
cp	0	0	1	0	1	
bp	0	1	0	0	1	
a	1	0	0	0	0	

Reținînd primele șapte linii din codul **K**, se obține codul din tabelul 13.9.

Codul din tabelul 13.9 se bucură de proprietatea că fiecare linie care are 1 în coloana p este fie un implicant al funcției F, fie include un astfel de implicant. Pe de altă parte, fiecare implicant prim al funcției F corespunde unei linii din **K** sau se include într-o astfel de linie.

Tabelul 13.9

	a	d	c	d	p	
abp	1	1	0	0	1	
acp	1	0	1	0	1	
bc	0	1	1	0	0	
$bcdp$	0	1	1	1	1	= **K**
acd	1	0	1	1	0	
abd	1	1	0	1	0	
dp	0	0	0	1	1	

Definiție. Un cod care îndeplinește proprietatea precedentă se numește *cod generat de funcția F.*

Analizînd codul din tabelul 13.9, se observă că toți implicanții primi ai funcției, cu excepția implicantului ad, se regăsesc printre liniile codului **K**. Însă implicantul prim ad se include în monomul d corespunzător liniei dp. Se poate trage concluzia că în F se repetă a. Vom căuta să verificăm acest lucru prin dedublarea coloanei a.

Fie deci $a = x \oplus y$, unde x și y sînt coloanele care corespund variabilelor contopite. Dacă

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_7 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_7 \end{pmatrix},$$

atunci trebuie să existe relațiile

$$\begin{aligned} x_1 \oplus y_1 &= 1, & x_5 \oplus y_5 &= 1, \\ x_2 \oplus y_2 &= 1, & x_6 \oplus y_6 &= 1, \\ x_3 \oplus y_3 &= 0, & x_7 \oplus y_7 &= 1 \\ x_4 \oplus y_4 &= 0, \end{aligned}$$

și în plus din relațiile $x_{2^s+t} = x_{2^s} \oplus x_t$ și $y_{2^s+t} = y_{2^s} \oplus y_t$, $s = 1, 2, t < 2^s$, se obțin relațiile

$$\begin{aligned} x_1 \oplus x_2 &= x_3, & y_1 \oplus y_2 &= y_3, \\ x_1 \oplus x_4 &= x_5, & y_1 \oplus y_4 &= y_5, \\ x_2 \oplus x_4 &= x_6, & y_2 \oplus y_4 &= y_6, \\ x_3 \oplus x_4 &= x_7, & y_3 \oplus y_4 &= y_7. \end{aligned}$$

Deoarece $x_4 \oplus y_4 = 0$ și implicantul bcd nu conține pe a, rezultă $x_4 = y_4 = 0$. La fel trebuie să admitem că $x_7 = y_7 = 1$, deoarece monomul d trebuie completat cu variabila a pentru a deveni implicant prim al funcției F. Mai departe, utilizînd aceste rezultate și relațiile de mai înainte, se obțin :

$$x_3 = x_7 = 1, \; y_3 = y_7 = 1.$$

Apoi, luînd arbitrară una dintre variabile, de exemplu x_1, rezultă :

a) $x_1 = 1 \Rightarrow y_1 = 1 \oplus x_1 = 0, \quad x_2 = x_1 \oplus x_3 = 0,$
$y_2 = 1 \oplus x_2 = 1, \quad x_5 = x_1 \oplus x_4 = 1,$
$y_5 = x_5 \oplus 1 = 0, \quad x_6 = x_2 \oplus x_4 = 0,$
$y_6 = x_6 \oplus 1 = 1,$

de unde

$$\mathbf{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

Înlocuind în codul **K'** coloana **a** cu coloanele **x**, **y**, se obține codul din tabelul 13.10, care este asociat funcției

$$F_1 = xb + yc + bcd + xyd.$$

Tabelul 13.10

	x	y	b	c	d	p	
xbp	1	0	1	0	0	1	
ycp	0	1	0	1	0	1	
$xybc$	1	1	1	1	0	0	
$bcdp$	0	0	1	1	1	1	$= \mathbf{K}$
xcd	1	0	0	1	1	0	
ybd	0	1	1	0	1	0	
$xydp$	1	1	0	0	1	1	

Această funcție se realizează prin schema din fig. 13.3, după cum se poate verifica ușor. Dacă egalăm pe x și y cu

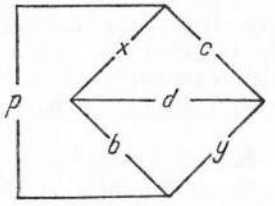

Fig. 13.3

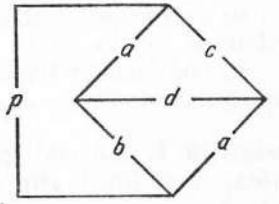

Fig. 13.4

a și înlocuim în fig. 13.3, se va obține schema din fig. 13.4 care realizează funcția

$$F = ab + ac + ad + bcd,$$

adică tocmai funcția din exemplul de față.

b) Dacă am fi luat $x_1 = 0$, s-ar fi permutat x cu y, obținîndu-se funcția

$$F_2 = yb + xc + bcd + xyd,$$

care conduce la aceeași soluție dacă egalăm pe x cu y.

13.2. Alte proprietăți ale codurilor grupale asociate funcțiilor booleene

Ne vom ocupa mai întîi de o proprietate folosită deja în subcapitolul precedent.

Proprietatea 1. *Într-un cod grupal oarecare*

$$\mathbf{K} = (\alpha_{ij}), \quad i = 1, 2, \ldots, 2^n, j = 1, 2, \ldots, n,$$

se poate realiza o ordonare a liniilor astfel încît să existe proprietatea A:

$$\alpha_{2^s+t} = \alpha_{2^s} \oplus \alpha_t, \quad s = 0, 1, 2, \ldots, m-1, \quad t < 2^s.$$

Demonstrație. Fie $a_1, a_2, \ldots, a_m$ o bază a codului **K**. Să luăm $\alpha_1 = a_1$ și $\alpha_2 = a_2$ și să notăm $\alpha_3 = \alpha_1 \oplus \alpha_2$. Atunci evident pentru $s = 1$, $t = 1$ relația A se verifică. În general, dacă luăm $\alpha_{2^s} = a_{s+1}$, $s = 0,1,2,\ldots, m-1$, și dacă notăm cu α_p, $2^s < p < 2^{s+1}$, linia $\alpha_{2^s} \oplus \alpha_t$, $t = p - 2^s$, atunci se obține ordonarea dorită. Într-adevăr, prin acest procedeu se obțin toate liniile codului **K**, întrucît a_1, $a_2, \ldots, a_m$ constituie o bază a codului respectiv, iar relațiile A rezultă din modul de construcție a liniilor lui **K**.

Definiția 1. Un cod grupal **K** este scris sub *formă canonică*, dacă liniile sînt ordonate astfel încît se verifică relațiile A.

Fie F o funcție booleană și **K** codul grupal generat de F. Dacă $F = \bigcup_i A_i$, $i \in \mathbf{I}$, este forma redusă a funcției F, unde **I** este o mulțime de indici, și dacă α_i, $i \in \mathbf{I}$, sînt liniile codului **K** corespunzătoare implicanților primi A_i ai lui F, atunci orice linie din **K** este combinație liniară a liniilor α_i, $i \in \mathbf{I}$, astfel că F determină univoc pe **K**.

Fie l coloana din **K** corespunzătoare laturii sursei, $\mathfrak{A} = \{\alpha_i\}$, $i \in \mathbf{I}$, mulțimea tuturor liniilor din **K** pentru care $\alpha_{il} = 1$ și b_j, $j = 1, 2, \ldots, l-1$, variabila booleană corespunzătoare coloanei j. Fie produsele

$$A_i = \prod_{j=1}^{l-1} b_j^{\alpha_{ij}}, \quad i \in \mathbf{I},$$

unde

$$\alpha_{ij} = 1 \Rightarrow b_j^{\alpha_{ij}} = b_j,$$

$$\alpha_{ij} = 0 \Rightarrow b_j^{\alpha_{ij}} = 1.$$

Am văzut că pentru funcțiile booleene care se realizează prin scheme fără repetiție și un număr de alte funcții booleene care se realizează prin scheme cu-repetiție, există relația $F = \bigcup_i A_i$, $i \in \mathbf{I}$, disjuncția fiind extinsă la toți A_i corespunzători liniilor $\alpha_i \in \mathfrak{A}$, în timp ce pentru alte funcții care se realizează prin scheme cu repetiție relația de mai sus nu se verifică.

Vom analiza în continuare relațiile care există între funcțiile booleene, codurile grupale generate de acestea și funcțiile care le realizează.

Definiția 2. Fiind dat un cod grupal **K**, vom numi *funcție booleană generată* de **K**, referitoare la coloana p, funcția $F_p = \bigcup_i A_i$, $i \in \mathbf{I}$, unde $A_i = \prod_{j \neq p} b_j^{\alpha_{ij}}$, $i \in \mathbf{I}$, iar **I** este mulțimea tuturor indicilor pentru care $\alpha_{ip} = 1$.

Exemplul 1. Pentru codul din tabelul 13.10 avem:

$F_1 = bp + cd + ybc + ydp$, $\quad F_4 = yp + xd + xyb + bdp$,

$F_2 = cp + bd + xbc + xdp$, $\quad F_5 = xc + yb + bcp + xyp$,

$F_3 = xp + yd + xyc + cdp$, $\quad F_6 = xb + yc + bcd + xyd$,

unde $F_1 = F_x$, $F_2 = F_y$, $F_3 = F_b$, $F_4 = F_c$, $F_5 = F_d$ și $F_6 = F_p$.

Un cod grupal **K** generează l funcții booleene, unde l este numărul de coloane din **K**.

Definiția 3. Fie F_0 funcție booleană și **K** codul grupal generat de F. Fie p indicele coloanei corespunzătoare

laturii sursei și F_p funcția booleană generată de **K** referitor la p. Prin definiție codul **K** este *caracteristic* pentru funcția F dacă există relația $F = F_p$.

Reciproc, vom putea introduce următoarea definiție.

Definiția 4. Fie **K** un cod grupal și F_p funcția booleană generată de **K** referitoare la coloana p. Fie $\mathbf{K}_p$ codul grupal generat de F_p. Prin definiție funcția booleană F_p este *caracteristică* pentru codul **K** dacă există egalitatea $\mathbf{K} = \mathbf{K}_p$, înțelegînd prin această egalitate că liniile codului **K** se regăsesc printre liniile codului $\mathbf{K}_p$ și invers.

Exemple de funcții cărora le corespund coduri caracteristice sînt funcțiile care se realizează prin scheme fără repetiție. Fie F o funcție booleană care se realizează prin scheme fără repetiție, **K** codul grupal generat de F și $\mathscr{S}_F$ multipolul care realizează funcția F. În acest caz unul oricare dintre cele trei elemente determină, în afara unor transformări de echivalență, în mod univoc pe celelalte două, după cum rezultă din algoritmii de sinteză dați în subcapitolele precedente.

În cazul funcțiilor care se realizează prin scheme cu repetiție lucrurile sînt diferite. Următorul exemplu ne demonstrează că existența codurilor și funcțiilor caracteristice nu este suficientă pentru determinarea univocă a schemelor care realizează funcțiile respective, adică pentru efectuarea sintezei.

Exemplul 2. Fie funcția booleană

$$F = axc + ayd + bxd + byc.$$

Această funcție generează codul **K** din tabelul 13.11.

Tabelul 13.11

	a	b	c	d	x	y	p	
$a\ x\ c\ p$	1	0	1	0	1	0	1	
$a\ y\ d\ p$	1	0	0	1	0	1	1	
$c\ d\ x\ y$	0	0	1	1	1	1	0	
$b\ x\ d\ p$	0	1	0	1	1	0	1	$= \mathbf{K}$
$a\ b\ c\ d$	1	1	1	1	0	0	0	
$a\ b\ x\ y$	1	1	0	0	1	1	0	
$b\ y\ c\ p$	0	1	1	0	0	1	1	

Funcțiile generate de codul **K** sînt următoarele :

$$F_1 = cxp + dyp + bcd + bxy, \qquad F_5 = acp + cdy + bdp + aby,$$
$$F_2 = dxp + acd + axy + cyp, \qquad F_6 = adp + cdx + abx + bcp,$$
$$F_3 = axp + dxy + abd + byp, \qquad F_7 = axc + ayd + bxd + byc,$$
$$F_4 = ayp + cxy + bxp + abc, \qquad \text{unde } F_7 \text{ este chiar } F.$$

Se poate verifica ușor că fiecare dintre cele șapte funcții generează codul **K**, și reciproc fiecare dintre aceste funcții se obține univoc din **K** folosind relația din definiția 2. Funcțiile F_i, $i = 1, 2, \ldots, 7$, și **K** sînt reciproc caracteristice.

După cum se știe, funcția F și implicit fiecare dintre funcțiile F_i nu se realizează prin scheme fără repetiție. În fig. 13.5 sînt date trei scheme, care realizează funcția F cu număr minim de contacte. Fără să ținem seama de transformări de echivalență, există deci trei scheme care realizează funcția F. Evident se pot obține și alte scheme dacă se pleacă de la altă funcție din mulțimea $\{F_1, F_2, \ldots, F_7\}$

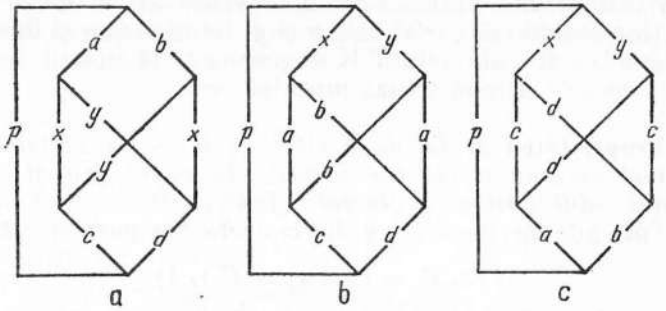

Fig. 13.5

de funcții, unde F_i, $i = 1, 2, \ldots, 7$, corespunde variabilei din coloana i.

Mai este de făcut următoarea observație : schemele din fig. 13.5 corespund numai funcției $F_p = F_7$ și nu corespund nici uneia dintre restul de șase funcții din exemplul 2. Într-adevăr, dacă în fig. 13.5 am considera latura a ca latură a sursei, s-ar obține funcția

$$F_a = bxy + bxcd + cxp + dyp$$

și se verifică ușor că $F_1 \neq F_a$, deoarece în F_a apare monomul *bxcd* în locul monomului *bcd* din F_1, unde F_1 este funcția corespunzătoare coloanei 1 din tabelul 13.11.

Dacă analizăm funcțiile din exemplul 1, vedem că toate acestea conduc la schema din fig. 13.3 și toate generează codul din tabelul 13.10.

Aceste observații privind relațiile dintre funcțiile booleene, codurile generate de acestea și funcțiile care le realizează sînt utile la elaborarea algoritmilor de sinteză și le vom folosi în continuare.

Proprietatea 2. *La două elemente a și b conectate în serie corespund coloane identice în codul* **K** *generat de funcția F.*

Într-adevăr, două variabile ale lui F sînt conectate în serie dacă

$$F(x, 0) = F(0, y) = F(0, 0),$$

adică $F = Axy + B$, unde A și B nu conțin pe x și pe y. Dar atunci un implicant prim oarecare A_i al lui F fie conține amîndouă variabilele x și y, fie nici una și deci în coloanele x și y ale codului **K** se găsește fie 1 în amîndouă, fie 0, ceea ce demonstrează proprietatea.

Proprietatea 3. *La două elemente a și b conectate în paralel corespund coloane identice în codul grupal* **K***, dualul codului grupal* **K** *generat de funcția F.*

Într-adevăr, dacă x și y sînt conectate în paralel, atunci

$$F(x, 1) = F(1, y) = F(1, 1)$$

și deci $F = A(x + y) + B$, unde A și B nu conțin pe x și y. Dar atunci

$$F^* = (A^* + xy)B^* = B^* xy + A^*B^*$$

și conform proprietății 2 în coloanele x și y ale codului **K*** apare simultan 1 sau 0.

Proprietatea 4. *Dacă funcțiilor booleene duale F și F* le corespund codurile caracteristice* **K** *și* **K***, atunci între aceste coduri există relația* $\mathbf{K}^*\mathbf{K}' = \Phi$, *denumită relație de ortogonalitate între coduri, unde* Φ *este o matrice de dimensiuni* $p \times q$ *cu toate elementele nule, codurile* **K*** *și* **K** *avînd dimensiunile* $p \times l$ *și respectiv* $q \times l$.

Într-adevăr, un element oarecare φ_{ik} al matricei Φ este dat de expresia

$$\sum_{j=1}^{l} \alpha_{ij}\, \beta_{kj}$$

și deoarece codurile sînt caracteristice, se verifică relația de ortogonalitate, ceea ce demonstrează proprietatea.

Definiția 4. Două coduri grupale care verifică relația de ortogonalitate între coduri se numesc *coduri duale.*

Rezultă imediat că dacă se dă un cod grupal **K**, se poate obține codul dual **K*** prin rezolvarea sistemului de ecuații booleene dat de relația de ortogonalitate în care elementele codului **K*** se consideră necunoscute.

Fie $\alpha_i = (\alpha_{i1}\, \alpha_{i2} \ldots \alpha_{il})$; $i = 1, 2, 3, \ldots, 2^m$, liniile codului **K** și $\mathbf{x} = (x_1, x_2, \ldots, x_l)$ un vector linie arbitrar. Rezolvarea sistemului de ortogonalitate se reduce la rezolvarea sistemului de ecuații

$$\sum_{j=1}^{l} \alpha_{ij} X_j = 0, \quad i = 1, 2, \ldots, 2^m,$$

unde X_j sînt necunoscute. La rîndul lui sistemul precedent se reduce la sistemul

$$\sum_{j=1}^{l} \alpha_{ij} X_j = 0, \quad i = 1, 2, \ldots, m,$$

denumit sistem redus sau sistem redus asociat codului **K**, unde s-a reținut numai un subsistem liniar independent de ecuații. Numărul de ecuații în sistemul redus este egal cu numărul m de linii liniar independente din **K**.

Sistemul redus se rezolvă cu ușurință prin metoda eliminării.

Exemplu. Să se afle codul dual **K*** al codului **K** din tabelul 13.10 prin rezolvarea sistemului redus asociat codului **K**.

Un sistem liniar independent de linii din **K** este dat de liniile

$$\alpha_1 = (1\ 0\ 1\ 0\ 0\ 1),$$
$$\alpha_2 = (0\ 1\ 0\ 1\ 0\ 1),$$
$$\alpha_4 = (0\ 0\ 1\ 1\ 1\ 1),$$

care conduce la sistemul

$$x \oplus b \oplus p = 0,$$
$$y \oplus c \oplus p = 0,$$
$$b \oplus c \oplus d \oplus p = 0,$$

din care se obține

$$x = b \oplus p,$$
$$y = c \oplus p,$$
$$d = b \oplus c \oplus p,$$

unde b, c, p sînt arbitrare. Se obține codul **K*** din tabelul 13.12, care conduce la funcția

$$F_p = xyd + xc + yb + bcd,$$

duala funcției

$$F = xb + yc + xyd + bcd,$$

funcția generatoare a codului **K** din tabelul 13.10.

Tabelul 13.12

	x	y	d	b	c	p	
$xydp$	1	1	1	0	0	1	
ydc	0	1	1	0	1	0	
xcp	1	0	0	0	1	1	
xdb	1	0	1	1	0	0	$= \mathbf{K}^*$
ybp	0	1	0	1	0	1	
$xybc$	1	1	0	1	1	0	
$bcdp$	0	0	1	1	1	1	

Am obținut astfel și o nouă metodă de construcție a funcției duale a unei funcții date. Această metodă prezintă avantajul că este ușor programabilă pe calculatorul electronic.

13.3. Funcții extinse ale funcțiilor booleene

Proprietățile codurilor grupale duale ne permit să extindem funcțiile booleene arbitrare la funcții booleene

cărora le corespund coduri caracteristice. Extinderea se face prin introducerea în funcțiile booleene inițiale a unor variabile suplimentare, în număr minim.

Fie $\{\alpha_{2^i-1}\}$; $i = 1, 2, \ldots, m$, un sistem de linii liniar independente ale codului **K** care constituie o bază în **K**. Vom asocia fiecărei linii numărul α_{2^m-i} astfel încît liniei α_1 îi corespunde numărul 2^{m-1}, iar liniei α_{2^m-1} numărul 1.

Definiția 1. Numim *rang* al coloanei j în codul grupal **K** numărul ν_j dat de expresia

$$\nu_j = \sum_{i=1}^{m} \alpha_{2^i-1_j}\, 2^{m-i}.$$

Similar vom introduce pentru coloana j a codului grupal **K***, dualul lui **K**, rangul

$$\mu_j = \sum_{k=1}^{n} \beta_{2^k-1_j}\, 2^{n-k},$$

unde m și n reprezintă numărul maxim de linii liniar independente în **K** și respectiv **K***. Rangurile ν_j, $j = 1, 2, \ldots, l$, reprezintă sume de puteri diferite ale lui 2. Dacă $\nu_j = \sum_{s=1}^{p_j} 2^{j_s}$ și $\nu_k = \sum_{r=1}^{p_k} 2^{k_r}$ sînt rangurile coloanelor j și k și dacă $\mathbf{M} = \{j_s\}$, $s = 1, 2, \ldots, p_j$, este mulțimea puterilor lui 2 în care se descompune ν_j iar $\mathbf{N} = \{k_r\}$, $r = 1, 2, \ldots, p_k$, este mulțimea puterilor lui 2 în care se descompune ν_k, între ν_j și ν_k vom introduce operația $\oplus$ definită astfel:

$$\nu_x = \nu_j \oplus \nu_k = \sum_{t=1}^{p_x} 2^{x_t}, \text{ unde mulțimea } \mathbf{Q} = \{x_t\},$$

$t = 1, 2, \ldots, p_x$, este dată de expresia $\mathbf{Q} = \mathbf{M} \cup \mathbf{N} - \mathbf{M} \cap \mathbf{N}$. În general, vom nota $\nu_1 \oplus \nu_2 \oplus \ldots \oplus \nu_n = \Xi_{i=1}^{n} \nu_i$.

Teorema 1. *Între rangurile* ν_j, $j = 1, 2, \ldots, l$, *și liniile codului* **K*** *există relațiile de ortogonalitate a rangurilor*

$$\sum_{j=1}^{l} \beta_{kj}\nu_j = 0, \quad k = 1, 2, \ldots, 2^n,$$

$$\sum_{j=1}^{l} \alpha_{ij}\mu_j = 0, \quad i = 1, 2, \ldots, 2^m.$$

Demonstrație. Vom demonstra mai întîi relațiile de ortogonalitate a rangurilor pentru ν_j. Înlocuind ν_j cu expresia corespunzătoare, vom obține succesiv :

$$\sum_{j=1}^{l} \beta_{kj} \sum_{i=1}^{m} \alpha_{2^{i-1}j}\, 2^{m-i} = \sum_{i=1}^{m} 2^{m-i} \sum_{j=1}^{l} \alpha_{2^{i-1}j}\, \beta_{kj} = 0,$$

întrucît conform relației de ortogonalitate între linii avem

$$\sum_{j=1}^{l} \alpha_{2^{i-1}j}\, \beta_{kj} = 0.$$

Deoarece linia β_k a codului **K*** a fost aleasă arbitrar, rezultă valabilitatea relațiilor de ortogonalitate a rangurilor pentru ν_j. La fel se demonstrează și relațiile de ortogonalitate a rangurilor pentru μ_j.

Teorema 2. *O condiție necesară și suficientă pentru ca două coduri* **K** *și* **K*** *să fie duale este ca rangurile* ν_j *sau ca rangurile* μ_j *să verifice relațiile de ortogonalitate.*

Demonstrație. Este suficient să demonstrăm echivalența relațiilor de ortogonalitate a rangurilor cu relațiile de ortogonalitate a liniilor codurilor. Faptul că relațiile de ortogonalitate a liniilor implică relațiile de ortogonalitate a rangurilor rezultă din teorema 1.

Implicația inversă rezultă din relațiile următoare, echivalente cu relațiile de ortogonalitate a rangurilor. Într-adevăr,

$$\left(\sum_{i=1}^{m} 2^{m-i} \sum_{j=1}^{l} \alpha_{2^{i-1}j}\, \beta_{kj} = 0 \right) \Rightarrow \sum_{j=1}^{l} \alpha_{2^{i-1}j}\, \beta_{kj} = 0,$$

adică toți coeficienții puterilor 2^{m-i} trebuie să fie nuli.

Exemplu. Să se verifice relațiile de ortogonalitate pentru codurile grupale duale din tabelul 13.10 și respectiv din tabelul 13.12.

Considerînd liniile

$$\alpha_1 = (1\ 0\ 1\ 0\ 0\ 1),$$

$$\alpha_2 = (0\ 1\ 0\ 1\ 0\ 1),$$

$$\alpha_4 = (0\ 0\ 1\ 1\ 1\ 1)$$

din codul **K** (tabelul 13.10), se obține rangurile

$$\nu_x = 4, \quad \nu_y = 2, \quad \nu_b = 5, \quad \nu_c = 3, \quad \nu_d = 1, \quad \nu_p = 7$$

și înlocuind în relațiile de ortogonalitate, se obțin :

$$\nu_x \oplus \nu_y \oplus \nu_d \oplus \nu_p = 4 \oplus 2 \oplus 1 \oplus 7 = 0,$$

$$\nu_y \oplus \nu_d \oplus \nu_c = 2 \oplus 1 \oplus 3 = 0,$$

$$\nu_x \oplus \nu_c \oplus \nu_p = 4 \oplus 3 \oplus 7 = 0,$$

$$\nu_x \oplus \nu_d \oplus \nu_b = 4 \oplus 1 \oplus 5 = 0,$$

$$\nu_y \oplus \nu_b \oplus \nu_p = 2 \oplus 5 \oplus 7 = 0,$$

$$\nu_x \oplus \nu_y \oplus \nu_b \oplus \nu_c = 4 \oplus 2 \oplus 5 \oplus 3 = 0,$$

$$\nu_d \oplus \nu_b \oplus \nu_c \oplus \nu_p = 1 \oplus 5 \oplus 3 \oplus 7 = 0.$$

La fel se obțin relațiile de ortogonalitate calculînd rangurile ν_x, ν_y, ν_b, ν_c, ν_d, ν_p ale coloanelor din codul **K*** (tabelul 13.12) și apoi înlocuind aceste ranguri în relațiile corespunzătoare liniilor codului **K** (tabelul 13.10).

Relațiile de ortogonalitate vor putea fi folosite pentru depistarea funcțiilor, cărora nu le corespund coduri caracteristice.

Fie F o funcție booleană, F^* duala ei, **K** codul grupal generat de F și **K*** codul grupal generat de F^*.

Teorema 3. *O condiție necesară și suficientă pentru ca unei funcții booleene* F *să nu îi corespundă un cod caracteristic este ca între codurile* **K** *și* **K*** *generate de* F *și respectiv de* F^* *să nu existe relația de ortogonalitate a codurilor, adică să existe relația* $\mathbf{K^*K'} \neq \emptyset$, *denumită și relație de neortogonalitate.*

Demonstrație. Dacă p este latura sursei în **K** și dacă codul **K** nu este caracteristic pentru F, atunci F_p, funcția generată de **K** referitor la colcana p, va fi diferită de F. Dar

$$F_p \neq F \Rightarrow F_p^* \neq F^*.$$

În plus $F_p \geqslant F$. Există deci în **K** o linie care corespunde unui implicant prim din F_p^* și nu este implicant prim și în F^*. Dacă $\alpha \in \mathbf{K}$ este linia corespunzătoare, atunci există o linie β în **K***, astfel încît $\alpha\beta \neq 0$, ceea ce demonstrează teorema.

În general, între liniile codurilor grupale generate de funcțiile booleene și rangurile coloanelor codurilor respective există relațiile de neortogonalitate :

$$\sum_{j=1}^{l} \beta_{kj}\nu_j = p_k, \qquad k = 1, 2, \ldots, 2^n,$$

$$\sum_{j=1}^{l} \alpha_{ij}\mu_j = q_i, \qquad i = 1, 2, \ldots, 2^m,$$

unde $p_k \neq 0$, $q_i \neq 0$ cel puțin pentru un i și respectiv un k.

Dacă funcțiilor F și F^* li se asociază coduri caracteristice, atunci $p_k = q_i = 0$, $(\forall)$ $i = 1, 2, \ldots, 2^m$; $k = 1, 2, \ldots, 2^n$, în care caz se verifică relațiile de ortogonalitate.

Existența unor numere $p_k \neq 0$ și $q_i \neq 0$ arată că funcția F nu se realizează prin scheme fără repetiție. Se pune deci problema adăugării unor variabile care se realizează prin scheme fără repetiție. Într-un prim pas vom transforma funcția F, și implicit pe F^*, într-o funcție căreia îi corespinde un cod caracteristic. Vom conveni să adăugăm variabile suplimentare în funcțiile F și F^*, pornind de la relațiile de neortogonalitate, după următorul algoritm.

Descrierea algoritmului. 1. Presupunem că plecăm de la relațiile de neortogonalitate referitoare la μ_j. Se consideră toate numerele $q_i \neq 0$ și se asociază numărului q_i variabila booleană y_i cu rangul $\mu_{y_i} = q_i$. La două numere egale $q_i = q_j$ se asociază aceeași variabilă $y_i = y_j$. Evident din relațiile de neortogonalitate se pot lua în considerare numai m relații care corespund unui sistem liniar independent de linii din **K** și unor implicanți din F. Numărul maxim de variabile suplimentare ce se vor adăuga nu va depăși deci pe m. Apoi, dacă α_i este o linie a codului **K** pentru care $q_i \neq 0$, atunci la implicantul prim al funcției F, corespunzător liniei α_i se va adăuga variabila y_i.

2. Fie $y_1, y_2, \ldots, y_p$ variabilele suplimentare introduse în pasul 1.

Se caută relațiile de dependență liniară dintre coloanele corespunzătoare variabilelor $y_1, y_2, \ldots, y_p$ și coloanele corespunzătoare variabilelor inițiale din codul **K**, dacă astfel de relații există.

Observație. Calculele se pot face la fel de bine în mulțimea rangurilor, ținînd seama de legea de compoziție a rangurilor.

Dacă r este numărul maxim de relații liniar independente, $r < p$, atunci vor fi eliminate r dintre variabilele $y_1, \ldots, y_p$, rezolvînd cele r ecuații care exprimă relațiile de dependență liniară.

Fie $R_i(y_1, y_2, \ldots, y_p, b_1, b_2, \ldots, b_n) = 0$, $i = 1, 2, \ldots, r$, cele r ecuații care exprimă dependența liniară a variabilelor $y_1, y_2, \ldots, y_p$ de variabilele inițiale $b_1, b_2, \ldots, b_n$. Presupunem că dorim să eliminăm variabilele $y_1, y_2, \ldots, y_r$. Fie

$$y_i = f_i(y_{r+1}, y_{r+2}, \ldots, y_p, b_1, b_2, \ldots, b_n),$$
$$i = 1, 2, \ldots, r,$$

expresiile care dau pe y_i în raport cu $y_{r+1}, y_{r+2}, \ldots, y_p$ și $b_1, b_2, \ldots, b_n$.

Observație. Am folosit aceeași notație pentru coloanele codului **K** și variabilele funcției F, întrucît nu se naște confuzie.

Fie

$$\mu_{y_i} = \sum_{k=1}^{p} \delta_{ir} + k\mu_{y_{r+k}} \oplus \sum_{j=1}^{l} \lambda_{ij}\mu_{b_j}$$

expresia rangului lui y_i în funcție de rangurile lui y_{r+k}, $k = 1, 2, \ldots, p$, și b_j, $j = 1, 2, \ldots, n$. Vom înlocui în relațiile de neortogonalitate pe q_i prin μ_{y_i} și μ_j prin μ_{b_j} și vom calcula expresia

$$\sum_{j=1}^{l} \alpha_{ij}\mu_{b_j} \oplus \mu_{y_i}.$$

Presupunem că aceasta se scrie sub forma

$$\sum_{j=1}^{l} \gamma_{ij}\mu_{b_j} \oplus \sum_{k=r}^{p} \delta_{ir+k}\mu_{y_{r+k}},$$

unde $\gamma_{ij} = \alpha_{ij} \oplus \lambda_{ij}$. Implicantul prim din F corespunzător liniei α_i va fi egal cu produsul

$$\prod_{j=1}^{l} b_j^{\gamma_{ij}} y_{r+k}^{\delta_{ir+k}}.$$

Similar se transformă toți implicanții primi ai funcției F pentru care $q_i \neq 0$. După această transformare, din F se va obține o funcție care depinde și de $b_1, b_2, \ldots, b_n$, dar și de variabilele suplimentare.

3. Fie $\Phi(y_1, y_2, \ldots, y_p, b_1, b_2, \ldots, b_n)$ funcția obținută din F prin procedeul descris la punctele 1 și 2. Se calculează rangurile μ_{y_i}, $i = 1, 2, \ldots, p$, și μ_{b_j}, $\ldots, j=1, 2, \ldots, n$, ale variabilelor $y_1, y_2, \ldots, y_p$ și respectiv $b_1 b_2, \ldots, b_n$. Se înlocuiesc în expresiile de neortogonalitate calculate pentru funcția F^* rangurile μ_{b_j} astfel calculate. Dacă $P_k = \mu_{y_i}$, atunci la implicantul prim al funcției F^* care corespunde liniei k din codul $\mathbf{K}^*$ se va adăuga variabila y_i. Se aplică apoi pasul 2 înlocuind codul $\mathbf{K}$ și funcția F prin codul $\mathbf{K}^*$ și funcția F^*. În urma acestui proces, după eliminarea variabilelor suplimentare redondante se vor obține funcțiile $\Phi(y_1, y_2, \ldots, y_p, b_1, b_2, \ldots, b_n)$ și $\Phi^*(y_1, y_2, \ldots, y_p, b_1, b_2, \ldots, b_n)$, cărora le corespund coduri caracteristice. Eliminarea variabilelor suplimentare trebuie să se facă astfel încît să se obțină expresii cu număr minim de variabile pentru Φ și Φ^*. Acest criteriu se va folosi la alegerea funcțiilor care se elimină.

Definiție. Funcțiile Φ și Φ^* obținute prin algoritmul precedent vor fi numite *funcții extinse ale funcțiilor* F și respectiv F^*.

Exemplu. Să se extindă funcția

$$F = ag + bh + ci + adh + bdg + abi + ach + adi$$

prin adăugarea de variabile suplimentare pînă la o funcție căreia îi corespunde un cod caracteristic.

Duala funcției F este funcția

$$F^* = abc + ghi + abi + adhi + bcdg + acgh + adhc.$$

Vom reține din codul $\mathbf{K}^*$ generat de F^* numai un sistem minimal de linii liniar independente care corespund unor implicanți primi din F^* pe care

le vom transcrie în tabelul 13.13. Atribuind liniilor $\beta_1, \beta_2, \beta_4, \beta_8, \beta_{16}$ ponderile înscrise în dreapta tabelului, vor rezulta următoarele ranguri:

$\mu_a = 23$, $\mu_b = 20$, $\mu_c = 17$, $\mu_d = 2$, $\mu_g = 9$, $\mu_h = 11$, $\mu_i = 14$, $\mu_p = 31$.

De asemenea din codul $\mathbf{K}$ generat de funcția F vom reține numai un sistem liniar independent de linii, pe care le vom transcrie în tabelul 13.14.

Tabelul 13.13

	a	b	c	d	g	h	i	p	
β_1	1	1	1	0	0	0	0	1	16
β_2	0	0	0	0	1	1	1	1	8
β_4	1	1	0	0	0	0	1	1	4
β_8	1	0	0	1	0	1	1	1	2
β_{16}	1	0	1	0	1	1	0	1	1

Tabelul 13.14

	a	b	c	d	g	h	i	p
α_1	1	0	0	0	1	0	0	1
α_2	0	1	0	0	0	1	0	1
α_4	0	0	1	0	0	0	1	1
α_8	1	0	0	1	0	1	0	1
α_{16}	1	1	0	0	0	0	1	1
α_{32}	1	0	0	1	0	0	1	1

Scriind relațiile de neortogonalitate pentru liniile din tabelul 13.14, se vor obține:

$$\mu_a \oplus \mu_g \oplus \mu_p = 1, \qquad \mu_a \oplus \mu_d \oplus \mu_h \oplus \mu_p = 1,$$
$$\mu_b \oplus \mu_h \oplus \mu_p = 0, \qquad \mu_a \oplus \mu_b \oplus \mu_i \oplus \mu_p = 18,$$
$$\mu_c \oplus \mu_i \oplus \mu_p = 0, \qquad \mu_a \oplus \mu_d \oplus \mu_i \oplus \mu_p = 4.$$

1. Aplicînd primul pas al algoritmului, vom introduce trei variabile suplimentare x, y, z cu rangurile $\mu_x = 1$, $\mu_y = 18$ și $\mu_z = 4$. Introducînd variabilele x, y, z în implicanții primi ai funcției F, aceasta va căpăta forma

$$\Phi = agx + bh + ci + adhx + abiy + adiz + bdg + achy.$$

2. Trecînd la pasul 2 al algoritmului, se observă că între rangurile μ_x, μ_y, μ_z și μ_a există relația

$$\mu_x \oplus \mu_y \oplus \mu_z \oplus \mu_a = 0,$$

deci poate fi eliminată o variabilă. Presupunem că dorim să eliminăm pe z care apare numai în implicantul prim *adiz* al funcției Φ. Dacă în expresia $\mu_a \oplus \mu_d \oplus \mu_i \oplus \mu_z$ se înlocuiește μ_z cu valoarea obținută din ecuația de mai sus, se obține

$$\mu_a \oplus \mu_d \oplus \mu_i \oplus \mu_x \oplus \mu_y \oplus \mu_a = \mu_d \oplus \mu_i \oplus \mu_x \oplus \mu_y,$$

care conduce la noul implicant prim *dixy* al funcției Φ, care se scrie

$$\Phi = agx + bh + ci + adhx + abiy + dixy + bdg + achy.$$

3. Vom rescrie liniile $\alpha_1, \alpha_2, \alpha_4, \alpha_8, \alpha_{16}, \alpha_{32}$ ale codului **K**, pentru funcția Φ astfel obținută și le vom asocia ponderi ca în tabelul 13.15. Noile ranguri ale variabilelor $a, b, c, d, g, h, i, x, y, p$ vor fi

$$\mu_a = 38,\ \mu_b = 18,\ \mu_c = 8,\ \mu_d = 5,\ \mu_g = 32,\ \mu_h = 20,$$
$$\mu_i = 11,\ \mu_x = 37,\ \mu_y = 3,\ \mu_p = 63.$$

Tabelul 13.15

	a	b	c	d	g	h	i	x	y	p	
α_1	1	0	0	0	1	0	0	1	0	1	32
α_2	0	1	0	0	0	1	0	0	0	1	16
α_4	0	0	1	0	0	0	1	0	0	1	8
α_8	1	0	0	1	0	1	0	1	0	1	4
α_{16}	1	1	0	0	0	0	1	0	1	1	2
α_{32}	0	0	0	1	0	0	1	1	1	1	1

Calculînd expresiile de neortogonalitate folosind liniile codului **K*** și ținînd seama de rangurile lui x și y, vom completa implicanții primi ai funcției F^* cu aceste noi variabile și vom construi funcția Φ^*. Se vor obține relațiile:

$$\mu_a \oplus \mu_b \oplus \mu_c \oplus \mu_p = \mu_y.$$
$$\mu_g \oplus \mu_h \oplus \mu_i \oplus \mu_p = 0,$$
$$\mu_a \oplus \mu_b \oplus \mu_i \oplus \mu_p = 0,$$
$$\mu_a \oplus \mu_d \oplus \mu_h \oplus \mu_i \oplus \mu_p = \mu_y,$$
$$\mu_a \oplus \mu_c \oplus \mu_g \oplus \mu_h \oplus \mu_p = \mu_x,$$
$$\mu_b \oplus \mu_c \oplus \mu_d \oplus \mu_g \oplus \mu_p = 0,$$
$$\mu_a \oplus \mu_i \oplus \mu_d \oplus \mu_h \oplus \mu_p = 0.$$

Dar între rangurile variabilelor x, y și a există relația

$$\mu_x \oplus \mu_y \oplus \mu_a = 0,$$

deci mai poate fi eliminată o variabilă. Vom elimina variabila care conduce la numărul minim de variabile în Φ, adică a. Funcția Φ se va scrie după eliminarea lui a astfel:

$$\Phi = gy + bh + ci + dhy + bix + dixy + bdg + chx.$$

Funcția Φ, în afara simbolurilor variabilelor, este aceeași cu funcția F din exemplul din 12.3, care se realizează prin scheme fără repetiție.

13.4. Un algoritm de sinteză pentru funcțiile care se realizează prin scheme cu repetiție

Aici se va descrie un algoritm de construcție prin recurență a multipolilor care realizează o funcție booleană dată. Algoritmul se aplică atît pentru funcții care se realizează prin scheme fără repetiție, cît și celor ce se realizează prin scheme cu repetiție. De asemenea algoritmul poate fi aplicat atît funcțiilor cărora le corespund coduri caracteristice, cît și celor cărora nu le corespund astfel de coduri. Cu toate acestea, în general, vom prefera ca o funcție căreia nu-i corespunde un cod caracteristic să fie în prealabil extinsă pînă la o funcție cu cod caracteristic.

Vom pune în evidență mai întîi cîteva proprietăți ale rangurilor variabilelor booleene, utile pentru descrierea algoritmului.

Proprietatea 1. *La două sau mai multe variabile conectate în serie corespund ranguri egale.*

Proprietatea rezultă imediat din faptul că două variabile conectate în serie au coloane identice în codul generat de funcția booleană dată.

Proprietatea 2. *Dacă $\mu_1, \mu_2, \ldots, \mu_p$ sînt rangurile variabilelor $x_1, x_2, \ldots, x_p$ care constituie un nod al unui multipol dat, atunci există relația*

$$\overset{p}{\underset{i=1}{\Xi}}\ \mu_i = 0.$$

Proprietatea este adevărată după cum s-a arătat mai înainte pentru funcții care se realizează prin scheme fără repetiție. Această proprietate urmează să fie folosită la stabilirea unui criteriu de recunoaștere a variabilelor care se repetă.

Un nod al unui multipol va putea fi reprezentat fie ca mulțime a laturilor conectate la el

$$\{x_1, x_2, \ldots, x_p\},$$

fie ca mulțime a rangurilor acestor variabile

$$\{\mu_1, \mu_2, \ldots, \mu_p\}.$$

Vom prefera să folosim simultan cele două forme și vom scrie un nod ca o pereche de mulțimi, sub forma

$$\frac{\{\mu_1, \mu_2, \ldots, \mu_p\}}{\{x_1, x_2, \ldots, x_p\}}.$$

Uneori este mai comod să stabilim rangurile variabilelor plecînd de la implicanții primi ai funcției F, nu de la liniile codului **K**. Într-adevăr, dacă atribuim ponderi implicanților primi, mai precis unui sistem maximal de implicanți primi liniar independenți, se pot obține rangurile astfel.

Dacă $B_1, B_2, \ldots, B_p$ este un sistem maximal de implicanți primi ai funcției F de variabilele $x_1, x_2, \ldots, x_l$ și dacă asociem implicantului prim B_i ponderea 2^{p-i}, $i = 1, 2, \ldots, p$, atunci rangul variabilei x_k va fi dat de relația

$$\mu_k = \sum_{i=1}^{p} \alpha_i 2^{p-i}$$

denumită relația rangurilor, unde $\alpha_i = 1$ dacă $x_k \in B_k$ și $\alpha_i = 0$ dacă $x_k \notin B_k$.

Rangurile pot fi construite prin recurență.

Proprietatea 3. *Fie $\mu_k^{(n)}$ rangul variabilei x_k într-un sistem $\mathbf{S}_n$ de n implicanți primi ai funcției F și $\mu_k^{(n+1)}$ rangul aceleiași variabile în sistemul $\mathbf{S}_{n+1}$ obținut din $\mathbf{S}_n$ prin*

adăugarea unui nou implicant prim B_{n+1}. Între rangurile $\mu_k^{(n)}$ și $\mu_k^{(n+1)}$ există relația de recurență a rangurilor

$$\mu_k^{(n+1)} = 2\mu_k^{(n)} + \alpha_{n+1},$$

unde $\alpha_{n+1} = 1$ dacă $x_k \in B_{n+1}$ și $\alpha_{n+1} = 0$ în caz contrar.

Într-adevăr, folosind relația rangurilor, se obține succesiv

$$\mu_k^{(n+1)} = \sum_{i=1}^{n+1} \alpha_i 2^{n+1-i} = 2 \sum_{i=1}^{n} \alpha_i 2^{n-i} + \alpha_{n+1} = 2\mu_k^{(n)} + \alpha_{n+1}.$$

Pentru depistarea variabilelor care se repetă se vor folosi și următoarele proprietăți de legătură între mulțimea implicanților primi, mulțimea nodurilor și mulțimea variabilelor, valabile pentru funcțiile care se realizează prin scheme fără repetiție.

Proprietatea 4. *Fiecare nod conține cel mult două variabile conținute într-un implicant prim.*

Proprietatea 5. *Fiecare variabilă apare în do ă și numai în două noduri.*

Aceste proprietăți rezultă din faptul că implicanții primi reprezintă cicluri simple, care nu trec de două ori printr-un nod.

Pentru ușurarea exprimării vom folosi:

Definiția 1. Un *nod* se numește *binar* dacă conține numai două variabile.

Evident, variabilele care compun un nod binar sînt conectate în serie și au ranguri egale. Pentru n variabile conectate în serie există $n - 1$ noduri binare care le conțin, ordinea conectării acestor variabile nefiind esențială.

Definiția 2. Fie F și $\mathscr{S}_F$ o funcție booleană și un dipol care realizează funcția respectivă. Fie de asemenea TF funcția obținută din F prin transformarea T, unde T este o permutare a variabilelor și $\mathscr{S}_{TF}$ un dipol care realizează funcția TF. Dacă $F = TF$, vom spune că dipolii $\mathscr{S}_F$ și $\mathscr{S}_{TF}$ sînt echivalenți și că dipolul $\mathscr{S}_{TF}$ realizează și el funcția F.

Exemplul 1. Dipolii din fig. 13.5 sînt echivalenți și reprezintă realizări ale funcției

$$F = axc + ayd + bxd + byc.$$

Fie $\mathbf{S}_1 = \{B_1, B_2, \ldots, B_n\}$ un subsistem de implicanți primi ai funcției F. Un implicant prim B al lui F este o combinație liniară a implicanților primi din subsistemul $\mathbf{S}_1$, dacă se poate scrie sub forma

$$B = \overset{n}{\underset{i=1}{\Xi}} \mu_i B_i,$$

unde μ_i poate să ia valorile 0 sau 1.

Fie $\mathbf{S}_2 = \{C_1, C_2, \ldots, C_m\}$ mulțimea tuturor implicanților primi ai lui F care sînt combinații liniare ai implicanților primi din subsistemul $\mathbf{S}_1$ și

$$F_1 = \sum_{i=1}^{m} C_i$$

funcția booleană obținută ca sumă a implicanților primi din subsistemul $\mathbf{S}_2$. Evident $F_1 \leqslant F$.

Definiția 3. Funcția F_1 dată de expresia $F_1 = \sum_{i=1}^{m} C_i$ va fi numită *funcția generată de subsistemul de implicanți primi* $\mathbf{S}_1$.

Vom putea trece acum la *descrierea algoritmului*.

1. Se ia un subsistem $\mathbf{S}_1 = \{B_1, B_2, \ldots, B_n\}$ de implicanți primi ai funcției F astfel încît funcția F_1 generată de implicanții primi din sistemul $\mathbf{S}_1$ să se realizeze prin scheme fără repetiție. Se poate lua $n = 1$, însă vom prefera să luăm în sistemul $\mathbf{S}_1$ un număr maxim de implicanți primi care conduc la scheme fără repetiție pentru a scurta procesul de sinteză.

Observație. Implicanții primi se iau multiplicați cu o variabilă corespunzătoare laturii sursei.

Se construiește schema $\mathcal{M}_1$ care realizează funcția F_1.

Fie $\mathfrak{N} = \{N_1, N_2, \ldots, N_q\}$ mulțimea nodurilor schemei $\mathcal{M}_1$. Fiecare nod $N \in \mathfrak{N}$ se va scrie sub forma

$$N = \frac{\{\mu_1, \mu_2, \ldots, \mu_p\}}{\{x_1, x_2, \ldots, x_p\}},$$

unde la numitor este mulțimea variabilelor conectate la nodul respectiv, iar la numărător rangurile corespunzătoare.

2. Se adaugă un nou implicant prim B_{n+1} și se recalculează rangurile folosind relația de recurență a rangurilor. Fie $\lambda_1, \lambda_2, \ldots, \lambda_p$ rangurile variabilelor din nodul N după aplicarea pasului 3. Între λ_i și μ_i există relația

$$\lambda_i = 2\,\mu_i + \alpha_i,\ i = 1, 2, \ldots, p,$$

unde $\alpha_i = 1$ dacă $x_i \in B_{n+1}$ și $\alpha_i = 0$ în caz contrar. Se calculează expresia

$$E = \overset{p}{\underset{i=1}{\Xi}} \lambda_i$$

pentru fiecare nod $N \subset \mathfrak{N}$. Dacă $E = 0$, atunci nodul N rămîne nemodificat. Dacă $E = 1$, atunci la nodul N se adaugă un nou rang λ egal cu 1 și mulțimea rangurilor pentru nodul N devine $\{\lambda, \lambda_1, \lambda_2, \ldots, \lambda_p\}$. Rangului λ urmează să i se asocieze una dintre variabilele noi care apar în B_{n+1}, necuprinse în implicanții primi din sistemul $\mathbf{S}_1$.

Vom conveni să scriem nodul N provizoriu sub forma

$$N = \frac{\{\lambda, \lambda_1, \lambda_2, \ldots, \lambda_p\}}{\{-, x_1, x_2, \ldots, x_p\}}.$$

Liniuța marchează locul în care urmează să fie adăugată o variabilă nouă. Dacă în B_{n+1} apar r variabile noi, atunci la sistemul $\mathfrak{N}$ se adaugă $r - 1$ noduri binare cu rangurile variabilelor egale cu 1. Locurile variabilelor din aceste noduri vor fi de asemenea marcate cu liniuțe. Dacă nu apar liniuțe, se trece la pasul 5.

3. Se reduce la minimum numărul de liniuțe prin permutarea variabilelor din nodurile binare. Dacă numărul de liniuțe este egal cu dublul numărului de variabile noi din B_{n+1}, se completează liniuțele cu literele noi din B_{n+1}, respectînd proprietățile 4 și 5. Se trece la pasul 5.

4. Dacă nu se îndeplinește condiția impusă la pasul 3, se repetă unele variabile.

Aflarea variabilelor care se repetă se va face plecînd de la observația: variabilele care se repetă trebuie să se

găsească, pe de o parte, printre variabilele din B_{n+1}, iar, pe de altă parte, printre variabilele care apar de două ori în nodurile în care am introdus rangurile $\lambda = 1$; dacă $\mathbf{B}$ este această ultimă mulțime, atunci variabilele care se repetă se vor găsi în intersecția $\mathbf{B} \cap \mathbf{B}_{n+1}$, unde prin $\mathbf{B}_{n+1}$ am notat mulțimea variabilelor care apar în implicantul B_{n+1}. Dacă $\mathbf{B} \cap \mathbf{B}_{n+1} = \emptyset$, se permută variabilele din nodurile binare sau se înlocuiesc unele variabile cu variabilele suplimentare deja introduse în locul lor, pînă cînd $\mathbf{B} \cap \mathbf{B}_{n+1} \neq \emptyset$ sau dispar liniuțele.

Dacă x este o variabilă care se repetă, se introduce o nouă variabilă pe care o notăm cu α, se înlocuiește x cu α în B_{n+1}, se revine la vechile ranguri și se trece la pasul 2, ținînd seama că noul implicant prim este B_{n+1}, corectat prin adăugarea lui α.

5. Se repetă pașii 2 la 4 pentru toți implicanții primi ai funcției F, adăugînd pe B_{n+1} la $\mathbf{S}_1$, schimbînd pe n în $n + 1$ și schimbînd de asemenea notațiile.

Exemplul 2. Să se facă sinteza pentru funcția

$$F = axc + ayd + bxd + byc.$$

Înmulțind-o cu variabila corespunzătoare laturii sursei, se obține

$$F_p = axcp + aydp + bxdp + bxcp.$$

Pasul 1. Se ia sistemul $\mathbf{S}_1 = \{axcp\}$ compus dintr-un singur implicant prim. O schemă care realizează funcția F_1 generată de $\mathbf{S}_1$ are nodurile:

$$\mathfrak{N} = \left\{ \frac{(1,1)}{(p,a)}, \frac{(1,1)}{(a,x)}, \frac{(1,1)}{(x,c)}, \frac{(1,1)}{(c,p)} \right\}.$$

Pasul 2. Se adaugă implicantul prim *aydp* și se calculează noile ranguri. Mulțimea nodurilor va fi

$$\mathfrak{N} = \left\{ \frac{(3,3)}{(p,a)}, \frac{(1,3,2)}{(-,a,x)}, \frac{(2,2)}{(x,c)}, \frac{(1,2,3)}{(-,c,p)}, \frac{(1,1)}{(-,-)} \right\}.$$

Ultimul nod binar a apărut din cauză că implicantul prim *aydp* conține două variabile noi y și d.

Pasul 3. Condiția impusă de pasul 3 fiind îndeplinită, se completează liniuțele cu literele y, d, respectînd proprietățile 4 și 5. Noua configurație a mulțimii nodurilor va fi

$$\mathfrak{N} = \left\{ \frac{(3,3)}{(p,a)}, \frac{(1,3,2)}{(y,a,x)}, \frac{(2,2)}{(x,c)}, \frac{(1,2,3)}{(d,c,p)}, \frac{(1,1)}{(y,d)} \right\}.$$

Se repetă pașii algoritmului, începînd cu pasul 2.

Pasul 2. Se adaugă implicantul prim *bxdp* și se calculează noile ranguri. Mulțimea nodurilor va fi

$$\mathfrak{N} = \left\{ \frac{(1,7,6)}{(-,p,a)}, \frac{(1,2,6,5)}{(-,y,a,x)}, \frac{(1,5,4)}{(-,x,c)}, \frac{(3,4,7)}{(d,c,p)}, \frac{(1,2,3)}{(-,y,d)} \right\}.$$

Pasul 3. Nu se respectă condiția impusă în pasul 3, întrucît apar 4 liniuțe, deși nu apar variabile noi în implicantul prim *bxdp*. Rezultă că se repetă unele variabile. Se trece la pasul 4.

Pasul 4. Mulțimea variabilelor care apar de două ori în nodurile marcate cu liniuțe este $\mathbf{B} = \{a, y, x\}$, $\mathbf{B} \cap \mathbf{B}_{n+1} = \{x\}$. Rezultă că se repetă variabila x. Introducem variabila suplimentară α și îl înlocuim pe x în B_{n+1} cu α. Repetînd pasul 2, se va obține

$$\mathfrak{N} = \left\{ \frac{(1,7,6)}{(-,p,a)}, \frac{(2,6,4)}{(y,a,x)}, \frac{(4,4)}{(x,c)}, \frac{(3,4,7)}{(d,c,p)}, \frac{(1,2,3)}{(-,y,d)} \right\}.$$

Pasul 3. În $\mathfrak{N}$ apar două liniuțe și în $B_{n+1} = bxdp$ două variabile noi. Înlocuim liniuțele cu α și b, introducînd și un nou nod binar:

$$\mathfrak{N} = \left\{ \frac{(1,7,6)}{(\alpha,p,a)}, \frac{(2,6,4)}{(y,a,x)}, \frac{(4,4)}{(x,c)}, \frac{(3,4,7)}{(d,c,p)}, \frac{(1,2,3)}{(b,y,d)}, \frac{(1,1)}{(\alpha,b)} \right\}.$$

Pasul 5. Se adaugă B_{n+1} la $\mathbf{S}_1$, se schimbă n în $n + 1$ și se trece la pasul 2.

Pasul 2. Se adaugă implicantul prim *bycp* și se calculează noile ranguri. Mulțimea nodurilor va deveni

$$\mathfrak{N} = \left\{ \frac{(6,2,15,12)}{(-,\alpha,p,a)}, \frac{(1,5,12,8)}{(-,y,a,x)}, \frac{(1,8,9)}{(-,x,c)}, \frac{(1,9,15)}{(d,c,p)}, \frac{(3,5,6)}{(b,y,d)}, \frac{(1,2,3)}{(-,\alpha,b)} \right\}.$$

Pasul 3. Nu se respectă condiția de nerepetare a variabilelor. Se trece la pasul următor.

Pasul 4. Mulțimea variabilelor care apar de două ori în nodurile cu liniuță este

$$\mathbf{B} = \{\alpha, a, x\}, \ \mathbf{B} \cap \mathbf{B}_{n+1} = O.$$

Se permută α cu b care se găsesc într-un nod binar și se obține

$$\mathfrak{N} = \left\{ \frac{(3,15,12)}{(b,p,a)}, \frac{(1,5,12,8)}{(-,y,a,x)}, \frac{(1,8,9)}{(-,x,c)}, \frac{(6,9,15)}{(d,c,p)}, \frac{(1,2,5,6)}{(-,\alpha,y,d)}, \frac{(1,2,3)}{(-,\alpha,b)} \right\}.$$

Mulțimea **B** va fi următoarea :

$$\mathbf{B} = \{y, x, \alpha\}, \quad \mathbf{B} \cap \mathbf{B}_{n+1} = \{y\}.$$

Se repetă deci y . Introducem variabila suplimentară β. Înlocuim pe y în B_{n+1} cu β și repetăm pasul 2.

Pasul 2. Implicantul prim *bycp* se înlocuiește cu $b\,\beta\,cp$ și se recalculează rangurile. Se va obține

$$\mathfrak{R} = \left\{ \frac{(3, 15, 12)}{(b, p, a)}, \frac{(4, 12, 8)}{(y, a, x)}, \frac{(1, 8, 9)}{(-, x, c)}, \right.$$

$$\left. \frac{(6, 9, 15)}{(d, c, p)}, \frac{(2, 4, 6)}{(\alpha, y, d)}, \frac{(1, 2, 3)}{(-, \alpha, b)} \right\}.$$

Pasul 3. Se înlocuiește liniuța cu β și se obține

$$\mathfrak{R} = \left\{ \frac{(3, 15, 12)}{(b, p, a)}, \frac{(4, 12, 8)}{(y, a, x)}, \frac{(1, 8, 9)}{(\beta, x, c)}, \right.$$

$$\left. \frac{(6, 9, 15)}{(d, c, p)}, \frac{(2, 4, 6)}{(\alpha, y, d)}, \frac{(1, 2, 3)}{(\beta, \alpha, b)} \right\}.$$

Se obține deci schema din fig. 13.6, care după ce se înlocuiește α cu x și β cu y se reduce la schema din fig. 13.5, *a*), care realizează funcția F.

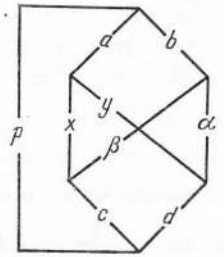

Fig. 13.6.

Pentru ilustrarea mai bună a algoritmului descris vom da un nou exemplu, mai complicat, care ne va conduce la sinteza sumei modulo 2 de n variabile.

Exemplul 3. Să se facă sinteza dipolului cu funcția de lucru :

$$F = adyl + acxl + beyl + bdxl + acyz + adxz + bdyz + bcxz.$$

Multiplicînd cu latura sursei p, vom obține expresia

$$F_p = adylp + acxlp + beylp + bdxlp + tacyzp + adxzp +$$

$$+ bdyzp + bcxzp.$$

Pasul 1. Se ia sistemul $\mathbf{S}_1$, format dintr-un singur implicant prim $\mathbf{S}_1 = \{adylp\}$, $F_1 = adylp$.

O schemă care realizează funcția F_1 are mulțimea de noduri

$$\mathfrak{R} = \left\{ \frac{(1, 1)}{(a, p)}, \frac{(1, 1)}{(a, y)}, \frac{(1, 1)}{(y, l)}, \frac{(1, 1)}{(l, d)}, \frac{(1, 1)}{(d, p)} \right\}.$$

Pasul 2. Adăugăm implicantul prim *acxlp* și calculăm noile ranguri. Se obține următoarea mulțime a nodurilor :

$$\mathfrak{R} = \left\{ \frac{(3, 3)}{(a, p)}, \frac{(1, 3, 2)}{(-, a, y)}, \frac{(1, 2, 3)}{(-, y, l)}, \frac{(1, 2, 3)}{(-, l, d)}, \frac{(1, 2, 3)}{(-, d, p)} \right\}.$$

Pasul 3. Reducem numărul de liniuțe prin permutarea variabilelor din nodurile binare. Permutînd l cu d, se obține mulțimea de noduri

$$\mathfrak{R} = \left\{ \frac{(3, 3)}{(a, p)}, \frac{(1, 3, 2)}{(-, a, y)}, \frac{(2, 2)}{(y, d)}, \frac{(1, 2, 3)}{(-, d, l)}, \frac{(3, 3)}{(l, p)} \right\}.$$

Deoarece în implicantul *acxlp* apar două variabile noi. trebuie să apară și un nod binar. În această situație mulțimea nodurilor după ce înlocuim liniuțele cu variabilele c și x, va fi

$$\mathfrak{R} = \left\{ \frac{(3, 3)}{(a, p)}, \frac{(1, 3, 2)}{(x, a, y)}, \frac{(2, 2)}{(y, d)}, \frac{(1, 2, 3)}{(c, d, l)}, \frac{(3, 3)}{(l, p)}, \frac{(1, 1)}{(c, x)} \right\}.$$

Pasul 5. Se trece la pasul 2.

Pasul 2. Se adaugă implicantul prim *beylp* și se obține mulțimea de noduri

$$\mathfrak{R} = \left\{ \frac{(1, 6, 7)}{(-, a, p)}, \frac{(1, 2, 6, 5)}{(-, x, a, y)}, \frac{(1, 5, 4)}{(-, y, d)}, \frac{(3, 4, 7)}{(c, d, l)}, \frac{(7, 7)}{(l, p)}, \frac{(1, 3, 2)}{(-, c, x)} \right\}.$$

Pasul 3. Nu se respectă condiția de nerepetare a variabilelor. Se trece la pasul următor.

Pasul 4. Mulțimea **B** a variabilelor care se pot repeta este

$$\mathbf{B} = \{a, x, y\}, \quad \mathbf{B} \cap \mathbf{B}_{n+1} = \{y\}.$$

Se repetă variabila y. Vom introduce variabila suplimentară și vom înlocui y cu α în *beylp*. Se trece la pasul 2.

Pasul 2. Se ia $B_{n+1} = bc\,\alpha\,lp$. Mulțimea de noduri recalculată va fi

$$\mathfrak{R} = \left\{ \frac{(1, 6, 7)}{(-, a, p)}, \frac{(2, 6, 4)}{(x, a, y)}, \frac{(4, 4)}{(y, d)}, \frac{(3, 4, 7)}{(c, d, l)}, \frac{(7, 7)}{(l, p)}, \frac{(1, 3, 2)}{(-, c, x)} \right\}.$$

Pasul 3. Adăugînd un nod binar și înlocuind pe b și α în locul liniuțelor, mulțimea nodurilor devine:

$$\mathfrak{N}_1 = \left\{ \frac{(1,6,7)}{(b,a,p)}, \frac{(2,6,4)}{(x,a,y)}, \frac{(4,4)}{(y,d)}, \frac{(3,4,7)}{(c,d,t)}, \frac{(7,7)}{(t,p)}, \frac{(1,3,2)}{(\alpha,c,x)}, \frac{(1,1)}{(b,\alpha)} \right\};$$

Pasul 5. Se trece la pasul 2.

Pasul 2. Se adaugă un nou implicant prim $B_{n+1} = bdxtp$

Mulțimea de noduri va deveni

$$\mathfrak{N} = \left\{ \frac{(3,12,15)}{(b,a,p)}, \frac{(1,5,12,8)}{(-,x,a,y)}, \frac{(1,8,9)}{(-,y,d)}, \frac{(6,9,15)}{(c,d,t)}, \frac{(15,15)}{(t,p)}, \frac{(1,2,6,5)}{(-,\alpha,c,x)}, \frac{(1,3,2)}{(-,b,\alpha)} \right\};$$

Pasul 3. Nu se respectă condiția de nerepetare a variabilelor; se trece la pasul următor.

Pasul 4. Mulțimea $\mathbf{B} = \{x, y, \alpha\}$; $\mathbf{B} \cap \mathbf{B}_{n+1} = \{x\}$.

Se repetă variabila x. Se introduce variabila suplimentară β și se înlocuiește x cu β în B_{n+1}. Se repetă pasul 2, luînd $B_{n+1} = bd\beta tp$.

Pasul 2. Se obține mulțimea de noduri

$$\mathfrak{N} = \left\{ \frac{(3,12,15)}{(b,a,p)}, \frac{(4,12,8)}{(x,a,y)}, \frac{(1,8,9)}{(-,y,d)}, \frac{(6,9,15)}{(c,d,t)}, \frac{(15,15)}{(t,p)}, \frac{(2,6,4)}{(\alpha,c,x)}, \frac{(1,3,2)}{(-,b,\alpha)} \right\}.$$

Pasul 3. Se înlocuiește în $\mathfrak{N}$ liniuța cu β și se obține

$$\mathfrak{N} = \left\{ \frac{(3,12,15)}{(b,a,p)}, \frac{(4,12,8)}{(x,a,y)}, \frac{(1,8,9)}{(\beta,y,d)}, \frac{(6,9,15)}{(c,d,t)}, \frac{(15,15)}{(t,p)}, \frac{(2,6,4)}{(\alpha,c,x)}, \frac{(1,3,2)}{(\beta,b,\alpha)} \right\}.$$

Pasul 5. Se trece la pasul 2.

Pasul 2. Se adaugă un nou implicant prim luînd $B_{n+1} = aeyzp$. Mulțimea nodurilor va deveni

$$\mathfrak{N} = \left\{ \frac{(6,25,31)}{(b,a,p)}, \frac{(8,25,17)}{(x,a,y)}, \frac{(1,2,17,18)}{(-,\beta,y,d)}, \frac{(1,13,18,30)}{(-,c,d,t)}, \frac{(1,30,31)}{(-,t,p)}, \frac{(1,4,13,8)}{(-,\alpha,c,x)}, \frac{(2,6,4)}{(\beta,b,\alpha)} \right\}.$$

Pasul 3. Nu se respectă condiția de nerepetare a variabilelor. Se trece la pasul următor.

Pasul 4. Mulțimea variabilelor care se pot repeta este

$$\mathbf{B} = \{d, c, t\}, \quad \mathbf{B} \cap \mathbf{B}_{n+1} = \{c\}.$$

Se repetă variabila c. Se introduce variabila suplimentară γ și se înlocuiește c cu γ în B_{n+1}. Se repetă pasul 2.

Pasul 2. $B_{n+1} = ac\gamma yzp$. Mulțimea nodurilor va deveni

$$\mathfrak{N} = \left\{ \frac{(6,25,31)}{(b,a,p)}, \frac{(8,25,17)}{(x,a,y)}, \frac{(1,2,17,18)}{(-,\beta,y,d)}, \frac{(12,18,30)}{(c,d,t)}, \frac{(1,30,31)}{(-,t,p)}, \frac{(4,12,8)}{(\alpha,c,x)}, \frac{(2,6,4)}{(\beta,b,\alpha)} \right\}.$$

Pasul 3. Se introduce un nod binar și se înlocuiesc liniuțele cu z și γ, după care se obține

$$\mathfrak{N} = \left\{ \frac{(6,25,31)}{(b,a,p)}, \frac{(8,25,17)}{(x,a,y)}, \frac{(1,2,17,18)}{(z,\beta,y,d)}, \frac{(12,18,30)}{(c,d,t)}, \frac{(1,30,31)}{(\gamma,t,p)}, \frac{(4,12,8)}{(\alpha,c,x)}, \frac{(2,6,4)}{(\beta,b,\alpha)}, \frac{(1,1)}{(z,\gamma)} \right\}.$$

Pasul 5. Se trece la pasul 2.

Pasul 2. Se adaugă implicantul prim $B_{n+1} = adxzp$ și se obține

$$\mathfrak{N} = \left\{ \frac{(12,51,63)}{(b,a,p)}, \frac{(17,51,31)}{(x,a,y)}, \frac{(3,4,34,37)}{(z,\beta,y,d)}, \frac{(1,24,37,60)}{(-,c,d,t)}, \frac{(1,2,60,63)}{(-,\gamma,t,p)}, \frac{(1,8,24,17)}{(-,\alpha,c,x)}, \frac{(4,12,8)}{(\beta,b,\alpha)}, \frac{(1,3,2)}{(-,z,\gamma)} \right\}.$$

Pasul 3. Nu se respectă condiția de repetare a variabilelor. Se trece la pasul următor.

Pasul 4. Mulțimea $\mathbf{B}$ este $\mathbf{B} = \{c, t, \gamma\}$, $\mathbf{B} \cap \mathbf{B}_{n+1} = \emptyset$.

Se permută z cu γ și se obține

$$\mathfrak{N} = \left\{ \frac{(12,51,63)}{(b,a,p)}, \frac{(1,7,51,34)}{(-,x,a,y)}, \frac{(1,2,4,31,37)}{(-,\gamma,\beta,y,d)}, \frac{(1,24,37,60)}{(-,c,d,t)}, \frac{(3,60,63)}{(z,t,p)}, \frac{(1,8,24,17)}{(-,\alpha,c,x)}, \frac{(4,12,8)}{(\beta,b,\alpha)}, \frac{(1,2,3)}{(-,z,\gamma)} \right\},$$

$$\mathbf{B} = \{\gamma, d, e\}, \mathbf{B} \cap \mathbf{B}_{n+1} = \{d\}.$$

Se repetă d. Se introduce variabila suplimentară δ, se înlocuiește d cu δ în B_{n+1} și se repetă pasul 2.

Pasul 2. $B_{n+1} = a\delta xzp$. Se obține

$$\mathfrak{R} = \left\{ \frac{(12, 51, 63)}{(b, a, p)}, \frac{(17, 51, 34)}{(x, a, y)}, \frac{(2, 4, 34, 36)}{(\gamma, \beta, y, d)}, \frac{(24, 36, 60)}{(c, d, t)}, \right.$$

$$\left. \frac{(3, 60, 63)}{(z, t, p)}, \frac{(1, 8, 24, 17)}{(-, \alpha, c, x)}, \frac{(4, 12, 8)}{(\beta, b, \alpha)}, \frac{(1, 2, 3)}{(-, z, \gamma)} \right\}.$$

Pasul 3. Se înlocuiește liniuța cu δ și $\mathfrak{R}$ devine

$$\mathfrak{R} = \left\{ \frac{(12, 51, 63)}{(b, a, p)}, \frac{(17, 51, 34)}{(x, a, y)}, \frac{(2, 4, 34, 36)}{(\gamma, \beta, y, d)}, \frac{(24, 36, 60)}{(c, d, t)}, \right.$$

$$\left. \frac{(3, 60, 63)}{(z, t, p)}, \frac{(1, 8, 24, 17)}{(\delta, \alpha, c, x)}, \frac{(4, 12, 8)}{(\beta, b, \alpha)}, \frac{(1, 2, 3)}{(\delta, z, \gamma)} \right\}.$$

Pasul 5. Se trece la pasul 2.

Pasul 2. Se ia $B_{n+1} = bdyzp$ și se obține

$$\mathfrak{R} = \left\{ \frac{(25, 102, 127)}{(b, a, p)}, \frac{(1, 34, 102, 69)}{(-, x, a, y)}, \frac{(4, 8, 69, 73)}{(\gamma, \beta, y, d)}, \frac{(1, 41, 73, 120)}{(-, c, d, t)}, \right.$$

$$\left. \frac{(7, 120, 127)}{(z, t, p)}, \frac{(12, 16, 48, 34)}{(\delta, \alpha, c, x)}, \frac{(1, 8, 25, 16)}{(-, \beta, b, \alpha)}, \frac{(1, 2, 4, 7)}{(-, \delta, \gamma, z)} \right\}.$$

Pasul 3. Nu se respectă condiția de nerepetare a variabilelor. Se trece la pasul 4.

Pasul 4. $\mathbf{B} = \emptyset$. Se verifică $y = \alpha$ și $z = \delta$. Se repetă pasul 2.

Pasul 2. $B_{n+1} = bd\alpha\delta p$. Se obține

$$\mathfrak{R} = \left\{ \frac{(25, 102, 127)}{(b, a, p)}, \frac{(34, 102, 68)}{(x, a, y)}, \frac{(4, 8, 68, 72)}{(\gamma, \beta, y, d)}, \frac{(48, 72, 120)}{(c, d, t)}, \right.$$

$$\left. \frac{(7, 120, 127)}{(z, t, p)}, \frac{(3, 17, 48, 34)}{(\delta, \alpha, c, x)}, \frac{(8, 25, 17)}{(\beta, b, \alpha)}, \frac{(3, 4, 7)}{(\delta, \gamma, z)} \right\}.$$

Nu apar liniuțe, deci se trece la pasul 5.

Pasul 5. Se trece la pasul 2 întrucît mai există implicanți primi neluați în considerare.

Pasul 2. $B_{n+1} = bcxzp$. Se obține

$$\mathfrak{R} = \left\{ \frac{(51, 204, 255)}{(b, a, p)}, \frac{(1, 69, 204, 136)}{(-, x, a, y)}, \frac{(8, 16, 136, 144)}{(\gamma, \beta, y, d)}, \right.$$

$$\frac{(1, 97, 144, 240)}{(-, c, d, t)}, \frac{(15, 240, 255)}{(z, t, p)}, \frac{(6, 34, 97, 69)}{(\delta, \alpha, c, x)},$$

$$\left. \frac{(1, 16, 41, 34)}{(-, \beta, b, \alpha)}, \frac{(1, 6, 8, 15)}{(-, \delta, \gamma, z)} \right\}.$$

Pasul 3. Nu se respectă condiția de nerepetare a variabilelor. Se trece la pasul următor.

Pasul 4. $\mathbf{B} = \emptyset$. Se verifică $c = \gamma$ și $x = \beta$ și se repetă pasul 2.

Pasul 2. $B_{n+1} = b\gamma\beta zp$. Se obține mulțimea de noduri

$$\mathfrak{R} = \left\{ \frac{(51, 204, 255)}{(b, a, p)}, \frac{(68, 204, 136)}{(x, a, y)}, \frac{(9, 17, 136, 144)}{(\gamma, \beta, y, d)}, \frac{(96, 144, 240)}{(c, d, t)}, \right.$$

$$\left. \frac{(15, 240, 255)}{(z, t, p)}, \frac{(6, 34, 96, 68)}{(\delta, \alpha, c, x)}, \frac{(17, 51,)}{(\beta, b, \alpha)}, \frac{(6, 9, 15)}{(\delta, \gamma, z)} \right\}.$$

Nu apar liniuțe, deci se trece la pasul 5.

Pasul 5. Procesul încetează, întrucît s-au epuizat implicanții primi. Se obține schema din fig. 13.7 care conduce la schema din fig. 13.8 prin înlocuirea variabilelor suplimentare cu variabilele inițiale corespunzătoare.

Schema din fig. 13.8 nu este singura care realizează funcția din exemplul 3. Se pot obține toate schemele prin permutarea între ele a perechilor de variabile (a, b), (x, y), (c, d), (t, z). Se obțin deci $4! = 24$ scheme distincte corespunzătoare tuturor transformărilor care lasă invariantă funcția F din exemplul respectiv.

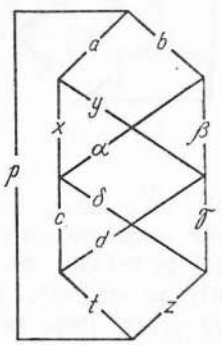

Fig. 13.7.

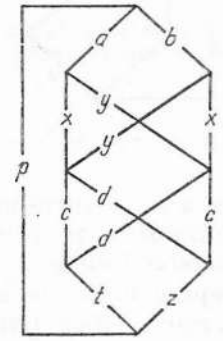

Fig. 13.8.

Dacă în exemplul 2 se înlocuiesc perechile de variabile (a, b), (x, y), (c, d) respectiv cu $(x_1, \bar{x}_1)$, $(x_2, \bar{x}_2)$, $(x_3, \bar{x}_3)$, funcția F din exemplul respectiv devine

$$F = x_1 \oplus x_2 \oplus x_3,$$

iar din fig. 13.5, *a*), făcînd înlocuirile corespunzătoare, se obține schema din fig. 13.9 care realizează funcția F.

La fel, dacă în exemplul 3 se înlocuiesc perechile de variabile (a, b), (x, y), (c, d), (t, z) respectiv cu $(x, \bar{x}_1)$, $(x_2, \bar{x}_2)$, $(x_3, \bar{x}_3)$, $(x_4, \bar{x}_4)$, se obține funcția

$$F = x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

și schema din fig. 13.10 care o realizează. În general, pentru funcția

$$F = x_1 \oplus x_2 \oplus \ldots \oplus x_m$$

se poate găsi prin același algoritm o schemă similară cu schemele din fig. 13.9 și 13.10, generalizarea făcîndu-se cu ușurință.

Observație. Pentru scurtarea procesului de sinteză este posibil să se lucreze numai cu implicanții primi liniar independenți.

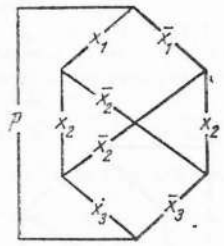

Fig. 13.9

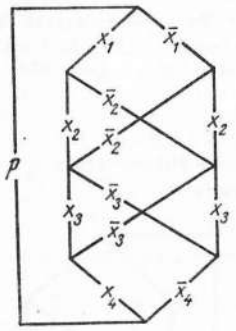

Fig. 13.10

Pentru a nu se lucra cu ranguri prea mari, în pasul 2 al algoritmului se poate reveni la rangurile inițiale dacă nu se introduc liniuțe. Implicanții primi care nu introduc liniuțe depind liniar de implicanții primi luați în considerare anterior și deci pot fi înlăturați. Este ceea ce se face prin revenire la variabilele inițiale.

Evident, algoritmul descris poate fi aplicat și funcțiilor care se realizează prin scheme fără repetiție.

În descrierea algoritmului am presupus că funcției care face obiectul sintezei îi corespunde un cod caracteristic. Întrucît în general acest lucru nu se întîmplă, în prealabil funcția trebuie extinsă pînă la o funcție căreia îi corespunde un cod caracteristic.

Fie $\Phi(x_1, x_2, \ldots, x_n; y_1, y_2, \ldots, y_m)$ funcția extinsă a funcției $F(x_1, x_2, \ldots, x_n)$, unde $y_1, y_2, \ldots, y_m$ sînt variabile suplimentare introduse prin algoritmul de extindere. Pentru terminarea procesului de sinteză este necesar să determinăm necunoscutele $y_1, y_2, \ldots, y_m$ în funcție de variabilele inițiale $x_1, x_2, \ldots, x_n$. Dar pentru aceasta este suficient să rezolvăm ecuația $\Phi = F$, adică $\Phi\bar{F} + \bar{\Phi}F = 0$, sau ecuația $\Phi^* = F^*$, adică $\Phi^* \bar{F}^* + \bar{\Phi}^* F^* = 0$, unde Φ^* și F^* sînt dualele funcțiilor Φ și F.

Vom da un exemplu simplu pentru ilustrarea algoritmului complet de sinteză.

Exemplul 4. Să se facă sinteza dipolului cu funcția de lucru

$$F = ac + bd + axd,$$

$$F_p = acp + bdp + axdp.$$

1. Calculul funcției extinse:

$$F^* = ab + cd + ad + bxc,$$

$$F_p^* = abp + cdp + adp + bxcp.$$

Asociind ponderi implicanților primi ai funcției F^* după regula: $abp \Leftrightarrow 1$, $cdp \Leftrightarrow 2$, $adp \Leftrightarrow 4$, $bxcp \Leftrightarrow 8$, se obțin rangurile variabilelor:

$$\mu_a = 5,\ \mu_b = 9,\ \mu_c = 10,\ \mu_d = 6,\ \mu_x = 8,\ \mu_p = 15.$$

Calculînd expresiile de neortogonalitate pentru funcția F, se obțin:

$$acp \Rightarrow \mu_a \oplus \mu_c \oplus \mu_p = 0,$$

$$bdp \Rightarrow \mu_b \oplus \mu_d \oplus \mu_p = 0,$$

$$axdp \Rightarrow \mu_a \oplus \mu_x \oplus \mu_d \oplus \mu_p = 4.$$

Se introduce o variabilă suplimentară cu rangul 4. Fie α această variabilă. Atunci funcția Φ extinsă va fi

$$\Phi(a, b, c, d, x; \alpha) = ac + bd + a\alpha xd + bx\alpha c,$$

implicantul prim suplimentar $bx\alpha c$ fiind combinația liniară a celorlalți implicanți primi. Ținînd seama că

$$\bar{F} = \bar{a}\bar{b} + \bar{c}\bar{b} + \bar{a}\bar{b} + \bar{b}\bar{x}\bar{c}$$

și

$$\bar{\Phi} = \bar{a}\bar{b} + \bar{c}\bar{d} + \bar{a}\bar{d}\bar{x} + \bar{a}\bar{d}\,\bar{\alpha} + \bar{b}\bar{c}\bar{a} + \bar{b}\bar{c}\,\bar{\alpha},$$

se obține ecuația următoare:

$$\Phi\,\overline{F} + \overline{\Phi}\,F = a\bar{b}\bar{c}dx\,\overline{\alpha} + \bar{a}bc\bar{d}x\,\alpha = 0$$

și, simplificînd cu x, se obține

$$a\bar{b}\bar{c}d\,\overline{\alpha} + \bar{a}bc\bar{d}\,\alpha = 0,$$

de unde

$$\alpha = \overline{\bar{a}bc\bar{d}} + a\bar{b}\bar{c}d = a + \bar{b} + \bar{c} + d.$$

Să facem acum sinteza funcției Φ folosind algoritmul respectiv.

Pasul 1. Luăm $\mathbf{S}_1 = \{acp\}$ care conduce la

$$\mathfrak{R} = \left\{\frac{(1,1)}{(a,c)}, \frac{(1,1)}{(c,p)}, \frac{(1,1)}{(p,a)}\right\}.$$

Pasul 2. Luăm $B_{n+1} = bdp$ și calculăm rangurile:

$$\mathfrak{R} = \left\{\frac{(2,2)}{(a,c)}, \frac{(1,2,3)}{(-,c,p)}, \frac{(1,3,2)}{(-,p,a)}, \frac{(1,1)}{(-,-)}\right\}.$$

Pasul 3. Se înlocuiesc liniuțele cu b și d și se obține

$$\mathfrak{R} = \left\{\frac{(2,2)}{(a,c)}, \frac{(1,2,3)}{(b,c,p)}, \frac{(1,3,2)}{(d,p,a)}, \frac{(1,1)}{(b,d)}\right\}.$$

Pasul 5. Se trece la pasul 2.

Pasul 2. $B_{n+1} = a\alpha xdp$. Se obține

$$\mathfrak{R} = \left\{\frac{(1,5,4)}{(-,a,c)}, \frac{(1,2,4,7)}{(-,b,c,p)}, \frac{(1,3,7,5)}{(-,d,p,a)}, \frac{(1,2,3)}{(-,b,d)}, \frac{(1,1)}{(-,-)}\right\}.$$

Pasul 3. Se reduce numărul de liniuțe prin permutarea variabilelor din nodurile binare, fie de exemplu a cu c, și se obține

$$\mathfrak{R} = \left\{\frac{(1,4,5)}{(-,c,a)}, \frac{(2,5,7)}{(b,a,p)}, \frac{(3,7,4)}{(d,p,c)}, \frac{(1,2,3)}{(-,b,d)}, \frac{(1,1)}{(-,-)}\right\}.$$

Se înlocuiesc liniuțele cu α și x; se obține

$$\mathfrak{R} = \left\{\frac{(1,4,5)}{(\alpha,c,a)}, \frac{(2,5,7)}{(b,a,p)}, \frac{(3,7,4)}{(d,p,c)}, \frac{(1,2,3)}{(x,b,d)}, \frac{(1,1)}{(\alpha,x)}\right\}.$$

Pasul 5. Se trece la pasul 2.

Pasul 2. $B_{n+1} = bx\alpha cp$. Se obține

$$\mathfrak{R} = \left\{\frac{(3,9,10)}{(\alpha,c,a)}, \frac{(5,10,15)}{(b,a,p)}, \frac{(6,15,9)}{(d,p,c)}, \frac{(3,5,6)}{(x,b,d)}, \frac{(3,3)}{(\alpha,x)}\right\}.$$

Se trece la pasul 5.

Pasul 5. Procesul încetează.

Funcția Φ se realizează prin schema fără repetiție din fig. 13.11. Dar α poate lua oricare din valorile a, $\bar{b}$, $\bar{c}$, d și se obțin patru soluții la care corespund schemele din fig. 13.12 — 13.15.

Acest exemplu simplu demonstrează necesitatea extinderii funcțiilor de lucru înainte de aplicarea algo-

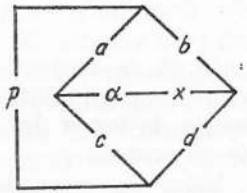

Fig. 13.11

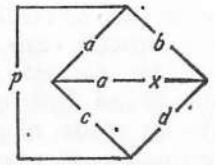

Fig. 13.12

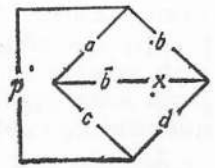

Fig. 13.13

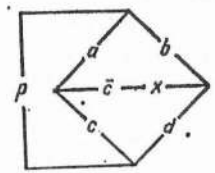

Fig. 13.14

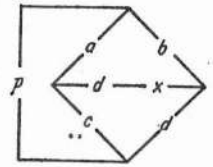

Fig. 13.15

ritmului de sinteză. Într-adevăr, funcția F din exemplul 4 se obține din funcția

$$F_1 = ac + bd + axd + bxc$$

prin înlăturarea implicantului prim bxc. Dar bxc este o combinație liniară a restului de implicanți primi. Dacă am

fi aplicat în mod mecanic algoritmul implicanților primi, am fi obținut schema din fig. 13.16, care de fapt realizează funcția F_1 și nu F, care se realizează prin scheme cu repetație.

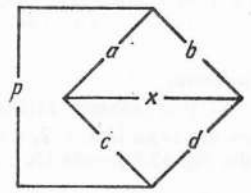

Fig. 13.16

Se obține astfel o proprietate foarte importantă și oarecum neprevăzută : prin eliminarea unor implicanți primi dintr-o funcție care se realizează prin scheme fără repetiție se obține o funcție care poate să se realizeze prin scheme cu repetiție. Cu alte cuvinte, prin înlăturarea unor implicanți primi dintr-o funcție de lucru dată numărul de contacte nu scade, ci poate să crească.

Aceasta este rațiunea pentru care luăm ca expresie minimă a unei funcții suma tuturor implicanților săi primi.

CAPITOLUL 14

ALGORITMUL GENERAL DE SINTEZĂ A AUTOMATELOR FINITE

14.1. Etapele sintezei automatelor finite

Să prezentăm succint dar complet metoda de sinteză care diferă de metodele clasice [3, 16, 35], bazate pe metoda lui Huffman, din două puncte de vedere : primul constă în comasarea unora dintre etapele sintezei, utilizate în metodele menționate, iar al doilea în adăugarea de noi etape. Aceste modificări au fost impuse, pe de o parte, de nevoia prelucrării automate a procesului de sinteză, iar pe de altă parte din dorința de a realiza o metodă completă de sinteză, care să cuprindă toate etapele și fazele, de la analiza problemei la realizarea fizică concretă a automatului care rezolvă problema dată.

După cum s-a arătat, un automat finit este un obiect $\mathcal{A}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \delta, \lambda)$ compus din trei mulțimi finite $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$, și două funcții δ și λ : mulțimea $\mathbf{X}$ a intrărilor, mulțimea $\mathbf{Y}$ a ieșirilor, mulțimea $\mathbf{Z}$ a stărilor, funcția reflexă δ : $\delta\ (\mathbf{X},\ \mathbf{Z}) \to \mathbf{Z}$ care aplică produsul $\mathbf{X} \times \mathbf{Z}$ în $\mathbf{Z}$ și funcția de transcodaj λ : $\lambda(\mathbf{X}, \mathbf{Z}) \to \mathbf{Y}$ care aplică produsul $\mathbf{X} \times \mathbf{Z}$ în $\mathbf{Y}$.

Schematic, automatul finit poate fi reprezentat ca un sistem cu reacție ca în fig. 14.1, unde față de elementele menționate am introdus și o funcție de întîrziere τ : $\tau(\mathbf{Z}) \to \mathbf{Z}$.

Din punct de vedere matematic automatul este complet definit dacă se dau mulțimea de intrare $\mathbf{X}$, mul-

țimea de ieșire **Y**, mulțimea stărilor **Z** și dacă se precizează funcțiile de tranziție δ și de ieșire λ. Din punct de vedere tehnic este important să se realizeze și schema sau schemele corespunzătoare celor două funcții de lucru. În plus este recomandabil ca realizarea automatului să fie cît mai eficientă din punct de vedere economic. Acest deziderat impune ca algoritmii folosiți să fie optimali în

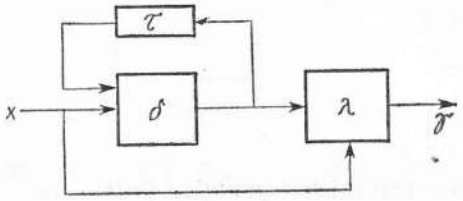

Fig. 14.1

sensul de a conduce la realizarea unor automate cu număr minim de elemente.

Nu am urmărit o optimizare globală, în schimb am căutat să obținem optimizări pe etape, algoritmii folosiți fiind optimali, în sensuri precizate de fiecare dată. Este interesant de văzut dacă optimele parțiale conduc sau nu la optimul global. Rămîne o problemă de rezolvat.

O preocupare majoră a fost problema automatizării procesului de sinteză.Pentru probleme simple sinteza se poate face la fel de ușor prin metodele intuitive ca și prin cele sistematice. Uneori soluțiile se pot găsi chiar mai ușor prin metodele intuitive întrucît folosesc scheme tip, standardizate, precum și experiența cîștigată în trecut. În schimb, problemele cu un grad mare de complexitate sînt practic inabordabile prin metodele intuitive. Este la fel de greu să rezolvi o problemă complexă printr-o metodă sistematică, prin calcule manuale. Soluția este deci naturală și constă în realizarea unor algoritmi programabili pe calculatorul electronic. Pornind de la această idee s-a urmărit, pe de o parte, adaptarea algoritmilor existenți la nevoile calculatorului, iar pe de altă parte, elaborarea unor algoritmi noi care să îndeplinească următoarele proprietăți de bază : să fie programabili, să

fie practici, adică să poată fi rezolvați în timp util pe calculator, și să fie înlănțuibili pentru a se mări gradul de automatizare a procesului de sinteză. Etapele sintezei automatelor finite sînt următoarele :

1. Analiza problemei în urma căreia se obține asignarea primară a stărilor.

2. Minimizarea numărului de stări ale automatului care conduce la matricea de tranziție a stărilor și la matricea ieșirilor.

3. Asignarea elementelor intermediare sau secundare cu respectarea condiției tehnice de bună funcționare.

4. Determinarea funcției de lucru și funcției de ieșiri ale automatului.

5. Simplificarea funcției de lucru și funcției de ieșiri.

6. Recunoașterea funcțiilor de lucru care se realizează prin scheme fără repetiție. Este un test care dacă se verifică, atunci se trece la etapa a șaptea, dacă nu, se trece la etapa a opta.

7. Sinteza multipolilor fără repetiție. Dacă funcțiile de lucru care au stat la baza sintezei sînt funcții extinse, se trece la etapa a 12-a, altfel sinteza este terminată și se trece la pasul 13.

8. Se verifică dacă codurile **K** și **K*** asociate funcțiilor F și F^* sînt ortogonale. Dacă da, se trece la etapa a noua, dacă nu, se trece la etapa a zecea.

9. Sinteza multipolilor cu repetiție. În urma acestei etape se obține o schemă a multipolului căutat în care apar variabilele inițiale alături de variabilele suplimentare, adăugate în procesul de sinteză. Se trece la pasul 11.

10. Construcția funcției de lucru extinse. În urma acestei etape se trece la pasul 6.

11. Scrierea funcției de lucru a multipolului obținut în pasul 9.

12. Rezolvarea ecuațiilor booleene sau ecuațiilor de variabile polivalente pentru identificarea variabilelor inițiale care corespund variabilelor suplimentare.

13. Terminarea procesului de sinteză.

Schema logică a procesului de sinteză este dată în fig. 14.2.

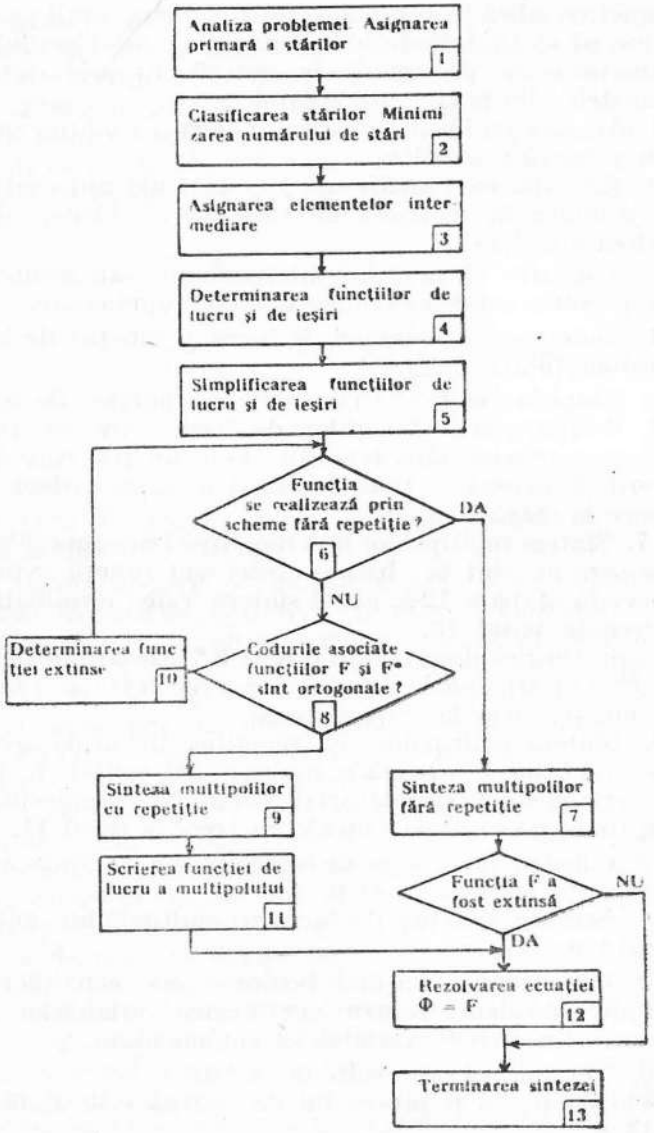

Fig. 14.2

14.2. Aplicarea algoritmului general de sinteză a automatelor finite

Pentru aplicarea algoritmului general de sinteză a automatelor vom alege ca exemplu sinteza registrului de decalare [35, 46].

Exemplu. Să se construiască un automat cu două variabile de intrare x_1 și x_2 și o variabilă de ieșire y, cu următoarele proprietăți:

a) la fiecare schimbare a valorii lui x_2 de la 0 la 1, variabila y ia valoarea lui x_1;

b) tranzițiile lui x_1 de la 1 la 0 sau de la 0 la 1 sînt fără influență asupra ieșirii;

c) tranzițiile lui x_2 de la 1 la 0 sînt fără efect asupra ieșirii;

d) schimbarea de stare simultană a lui x_1 și x_2 este exclusă.

1. *Analiza problemei. Asignarea primară a stărilor.* Există patru stări ale variabilelor de intrare {00 , 01, 11, 10} și două stări ale variabilei de ieșire {0, 1}. La fiecare stare x_1x_2 a variabilelor de intrare există posibilitatea să varieze x_1 sau x_2, dar nu x_1 și x_2 simultan. Atunci fiecărei stări x_1x_2 a variabilelor de intrare îi corespund două stări stabile ale automatului: una care corespunde valorii 0 a ieșirii și alta valorii 1 a ieșirii. Fie a una dintre cele două stări stabile ale automatului corespunzătoare stării x_1x_2 a variabilelor de intrare și a_1, a_2 stările automatului în care trece a la variația lui x_1 și respectiv x_2. Plecînd din starea a, se pot întîmpla următoarele cazuri:

A. Variază x_1 de la 1 la 0.
B. Variază x_1 de la 0 la 1.
C. Variază x_2 de la 1 la 0.
D. Variază x_2 de la 0 la 1.

Ținînd seama de proprietățile b) și c) ale automatului, rezultă că în cazurile A, B și C automatul nu are schimbare de stare la ieșire, deci a_1 și a_2 corespund aceleiași ieșiri y ca și a. În cazul D tranziția lui x_2 de la 0 la 1 va determina $y = x_1$ și dacă valoarea anterioară a lui y era $\bar{x}_1$

se produce o schimbare de stare la ieșire. În acest caz automatul trece din starea a cu ieșirea $\bar{x}_1$ într-o stare a_2 cu ieșirea x_1. O astfel de funcționare se poate reprezenta printr-un automat Moore cu asignarea primară a stărilor din tabelul 14.1.

Tabelul 14.1

Z \ x_1x_2	00	01	11	10	y
1	①	3	—	7	0
2	②	3	—	8	1
3	1	③	5	—	0
4	2	④	6	—	1
5	—	3	⑤	7	0
6	—	4	⑥	8	1
7	1	—	6	⑦	0
8	2	—	6	⑧	1

2. *Clasificarea stărilor. Minimizarea numărului de stări.* Pentru realizarea acestui pas se aplică algoritmul dat în cap. 10.

1°. Determinarea mulțimii de perechi de stări direct compatibile, $\mathfrak{M}$, mulțimii de perechi de stări direct incompatibile $\mathfrak{N}$ și mulțimii de prechi de stări nedeterminate $\mathfrak{P}$:

$\mathfrak{M} = \{(1, 3), (1, 5), (1, 7), (2,8), (3, 5), (4, 6), (4, 8), (6, 8)\}$,

$\mathfrak{N} = \{(1, 2), (1, 4), (1, 6), (1, 8), (2, 3), (2, 5), (2, 7), (3, 4), (3,6), (3,8), (4, 5), (4, 7), (5, 6), (5, 8), (6, 7), (7, 8)\}$,

$\mathfrak{P} = \{(2, 4), (2, 6), (3, 7), (5, 7)\}$.

2°. Determinarea mulțimilor de perechi de stări asociate perechilor de stări din $\mathfrak{P}$:

$$\mathbf{Z} = \{(\delta_{ik}, \delta_{jk}) \mid k \in \mathbf{K}_{ij}\},$$

unde $\mathbf{K}_{ij}$ este mulțimea acelor indici $k \in \{1, 2, 3, 4\}$ pentru care sînt definite amîndouă stările δ_{ik} și δ_{jk}.

$Z_{24} = \{(2, 2), (3, 4)\}$, $\quad Z_{26} = \{(3, 4), (8, 8)\}$,

$Z_{37} = \{(1, 1), (5, 6)\}$, $\quad Z_{57} = \{(5, 6), (7, 7)\}$.

3°. Scrierea expresiilor booleene asociate mulțimilor $\mathbf{Z}_{ij}$ și mulțimii $\mathbf{Z}$:

$$\mathcal{E}(\mathbf{Z}_{ij}) = e_{ij}, \quad (z_i, z_j) \in \mathfrak{M},$$
$$\mathcal{E}(\mathbf{Z}_{ij}) = \bar{e}_{ij}, \quad (z_i, z_j) \in \mathfrak{N},$$
$$\mathcal{E}(\mathbf{Z}_{24}) = e_{24}e_{34} + \bar{e}_{24}\bar{e}_{34},$$
$$\mathcal{E}(\mathbf{Z}_{26}) = e_{26}e_{34} + \bar{e}_{26}\bar{e}_{34},$$
$$\mathcal{E}(\mathbf{Z}_{37}) = e_{37}e_{56} + \bar{e}_{37}\bar{e}_{56},$$
$$\mathcal{E}(\mathbf{Z}_{57}) = e_{57}e_{56} + \bar{e}_{57}\bar{e}_{56}.$$

Observație. Determinarea mulțimilor $\mathbf{Z}_{ij}$ pentru perechile de stări direct compatibile și pentru cele direct incompatibile nu a fost necesară, întrucît nu intervin în definirea expresiilor booleene asociate.

$$\mathcal{E}(\mathbf{Z}) = e_{13}e_{15}e_{17}e_{28}e_{35}e_{46}e_{48}e_{68}\bar{e}_{12}\bar{e}_{14}\bar{e}_{16}\bar{e}_{18}\bar{e}_{23}\bar{e}_{25}\bar{e}_{27}\bar{e}_{34}\bar{e}_{36}\bar{e}_{38}$$
$$\bar{e}_{45}\bar{e}_{47}\bar{e}_{56}\bar{e}_{58}\bar{e}_{67}\bar{e}_{78}\,(e_{24}e_{34} + \bar{e}_{24}\bar{e}_{34})\,(e_{26}e_{34} + \bar{e}_{26}\bar{e}_{34})$$
$$(e_{37}e_{56} + \bar{e}_{37}\bar{e}_{56})(e_{57}e_{56} + \bar{e}_{57}\bar{e}_{56}) = e_{13}\bar{e}_{14}e_{15}\bar{e}_{16}e_{17}\bar{e}_{18}\bar{e}_{23}\bar{e}_{24}\bar{e}_{25}\bar{e}_{26}$$
$$\bar{e}_{27}e_{28}\bar{e}_{31}e_{35}\bar{e}_{36}\bar{e}_{37}\bar{e}_{38}\bar{e}_{45}\bar{e}_{46}\bar{e}_{47}e_{48}\bar{e}_{56}\bar{e}_{58}\bar{e}_{67}e_{68}\bar{e}_{78}.$$

Întrucît $\mathcal{E}(\mathbf{Z})$ are un singur constituent al unității, acesta este chiar elementul generator al automatului, $\mathcal{E}_A = \mathcal{E}(\mathbf{Z})$.

4. Scrierea matricei de compatibilitate $\mathfrak{A} = (\alpha_{ij})$, $i, j \in \{1, 2, \ldots, 8\}$, unde $\alpha_{ij} = 1$ dacă în $\mathcal{E}_A$ apare e_{ij} și $\alpha_{ij} = 0$ dacă în $\mathcal{E}_A$ apare $\bar{e}_{ij}$:

$$\mathfrak{A} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}.$$

Pe prima linie am înscris numerele stărilor corespunzătoare coloanelor din $\mathfrak{A}$.

Pentru condensarea scrierii o submatrice $\mathfrak{A}_X$, $\mathbf{X} \subset \mathbf{Z}$, a lui $\mathfrak{A}$ se va scrie sub forma $\mathfrak{A}_X = (\mathbf{X})$, unde în paranteză vom scrie elementele mulțimii $\mathbf{X}$. De exemplu, dacă $\mathbf{X} = \{3, 5, 8\}$, atunci vom scrie

$$\mathfrak{A}_X = \begin{matrix} 3 & 5 & 8 \\ \end{matrix} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (3, 5, 8).$$

Cu aceste notații $\mathfrak{A} = \mathfrak{A}_Z = (1, 2, 3, 4, 5, 6, 7, 8)$.

5°. Determinarea familiei submulțimilor maximale ale lui $\mathbf{Z}$. Se aplică algoritmul de determinare a submulțimilor maximale din 10.2.

Pasul 1. $\mathrm{T}_1^{(1)} = [(\{1, 2, 3, 4, 5, 6, 7, 8\}, \emptyset)]$, $T^{(1)} = \emptyset$.

Pasul 2. $\mathrm{T}_1^{(2)} = [(\{3, 5, 7\}, \{1\}), (\{2, 3, 4, 5, 6, 7, 8\}, \emptyset)]$,
$\mathrm{T}^{(2)} = \emptyset$.

Pasul 3. $\mathrm{T}_1^{(3)} = [(\{3, 5\}, \{1\}), (\{5, 7\}, \{1\}), (\{3, 8\}, \emptyset), (\{3, 4, 5, 6, 7, 8\}, \emptyset)]$,
$\mathrm{T}^{(3)} = [\{1, 3, 5\}, \{2, 8\}]$.

Pasul 4. $\mathrm{T}_1^{(4)} = [(\{5\}, \{1\}), (\{7\}, \{1\}), (\{3, 5\}, \emptyset), (\{4, 5, 6, 7, 8\}, \emptyset)]$,
$\mathrm{T}^{(4)} = [\{1, 3, 5\}, \{2, 8\}, \{1, 7\}]$.

Pasul 5. $\mathrm{T}_1^{(5)} = [(\{4, 6, 8\}, \emptyset), (\{5, 6, 7, 8\}, \emptyset)]$,
$\mathrm{T}^{(5)} = [\{1, 3, 5\}, \{2, 8\}, \{1, 7\}, \{4, 6, 8\}]$.

Pasul 6. $\mathrm{T}_1^{(6)} = [(\{5\}, \emptyset), (\{6, 7, 8\}, \emptyset)]$, $\mathrm{T}^{(6)} = \mathrm{T}^{(5)}$.

Pasul 7. $\mathrm{T}_1^{(7)} = [(\{6, 8\}, \emptyset), (\{7, 8\}, \emptyset)]$, $\mathrm{T}^{(7)} = \mathrm{T}^{(6)}$.

Pasul 8. $\mathrm{T}_1^{(8)} = [(\{7\}, \emptyset), (\{8\}, \emptyset)]$, $\mathrm{T}^{(8)} = \mathrm{T}^{(7)}$.

Pasul 9. $\mathrm{T}_1^{(9)} = \emptyset$, $\mathrm{T}^{(9)} = \mathrm{T}^{(8)}$.

Familia de submulțimi maximale în $\mathbf{Z}$ este următoarea :

$$\mathscr{Z}(\mathfrak{A}) = \{\{1, 3, 5\}, \{2, 8\}, \{1, 7\}, \{4, 6, 8\}\}.$$

6°. Stabilirea acoperirii minimale a mulțimii $\mathbf{Z}$ cu mulțimi din familia $\mathscr{Z}(\mathfrak{A})$. Asociem mulțimilor din $\mathscr{Z}(\mathfrak{A})$ variabilele booleene :

$A \leftrightarrow \{1, 3, 5\}$, $B \leftrightarrow \{2, 8\}$, $C \leftrightarrow \{1, 7\}$, $D \leftrightarrow \{4, 6, 8\}$.

Acoperirea minimală va fi dată de expresia booleană $(A + C)\ B\ A\ D\ A\ D\ C\ (B + D) = A\ B\ C\ D$. Rezultă că toate submulțimile din $\mathscr{Z}(\mathfrak{A})$ trebuie luate pentru a realiza o acoperire a lui $\mathbf{Z}$.

7°. Determinarea funcțiilor de tranziție și de ieșire. Notăm stările automatului redus astfel :

$$\mathbf{Z}_1 = \{1, 3, 5\},\ \mathbf{Z}_2 = \{2, 8\},\ \mathbf{Z}_3 = \{1, 7\},\ \mathbf{Z}_4 = \{4, 6, 8\}.$$

Se poate lua $Z_3 = \{7\}$.

Intrările vor fi notate astfel : $x^1 = 00$, $x^2 = 01$, $x^3 = 11$, $x^4 = 10$. Funcția de tranziție este următoarea :

$$\begin{aligned}
&\delta^*(\mathbf{Z}_1, x^1) = \{1\}, && \delta^*(\mathbf{Z}_1, x^2) = \{3\},\\
&\delta^*(\mathbf{Z}_1, x^3) = \{5\}, && \delta^*(\mathbf{Z}_1, x^4) = \{7\},\\
&\delta^*(\mathbf{Z}_2, x^1) = \{2\}, && \delta^*(\mathbf{Z}_2, x^2) = \{3\},\\
&\delta^*(\mathbf{Z}_2, x^3) = \{6\}, && \delta^*(\mathbf{Z}_2, x^4) = \{8\},\\
&\delta^*(\mathbf{Z}_3, x^1) = \{1\}, && \delta^*(\mathbf{Z}_3, x^2) = -,\\
&\delta^*(\mathbf{Z}_3, x^3) = \{6\}, && \delta^*(\mathbf{Z}_3, x^4) = \{7\},\\
&\delta^*(\mathbf{Z}_4, x^1) = \{2\}, && \delta^*(\mathbf{Z}_4, x^2) = \{4\},\\
&\delta^*(\mathbf{Z}_4, x^3) = \{6\}, && \delta^*(\mathbf{Z}_4, x^4) = \{8\}.
\end{aligned}$$

Fiecare $\delta^*(\mathbf{Z}_i, x^j)$, $i, j = 1, 2, 3, 4$, se include cel puțin într-o stare z_i, $i = 1, 2, 3, 4$. Există deci mai multe soluții. Una dintre ele are următoarea funcție de tranziție a stărilor :

$$\begin{aligned}
&\delta^*(\mathbf{Z}_1, x^1) = \mathbf{Z}_1, && \delta^*(\mathbf{Z}_1, x^2) = \mathbf{Z}_1, && \delta^*(Z_1, x^3) = \mathbf{Z}_1,\\
&\delta^*(\mathbf{Z}_1, x^4) = \mathbf{Z}_3, && \delta^*(\mathbf{Z}_2, x^1) = \mathbf{Z}_2, && \delta^*(\mathbf{Z}_2, x^2) = \mathbf{Z}_1,\\
&\delta^*(\mathbf{Z}_2, x^3) = \mathbf{Z}_4, && \delta^*(\mathbf{Z}_2, x^4) = \mathbf{Z}_4, && \delta^*(\mathbf{Z}_3, x^1) = \mathbf{Z}_1,\\
&\delta^*(\mathbf{Z}_3, x^2) = \mathbf{Z}_3, && \delta^*(\mathbf{Z}_3, x^3) = \mathbf{Z}_4, && \delta^*(\mathbf{Z}_3, x^4) = \mathbf{Z}_3,\\
&\delta^*(\mathbf{Z}_4, x^1) = \mathbf{Z}_2, && \delta^*(\mathbf{Z}_4, x^2) = \mathbf{Z}_4, && \delta^*(\mathbf{Z}_4, x^3) = \mathbf{Z}_4,\\
&\delta^*(\mathbf{Z}_4, x^4) = \mathbf{Z}_4.
\end{aligned}$$

Funcția de ieșire a automatului este următoarea :

$$\lambda^*(\mathbf{Z}_1, x^1) = \lambda(1, x^1) = 0, \qquad \lambda^*(\mathbf{Z}_1, x^2) = \lambda(3, x^2) = 0,$$
$$\lambda^*(\mathbf{Z}_1, x^3) = \lambda(5, x^3) = 0, \qquad \lambda^*(\mathbf{Z}_1, x^4) = \lambda(7, x^4) = 0,$$
$$\lambda^*(\mathbf{Z}_2, x^1) = \lambda(2, x^1) = 1, \qquad \lambda^*(\mathbf{Z}_2, x^2) = \lambda(3, x^2) = 0,$$
$$\lambda^*(\mathbf{Z}_2, x^3) = \lambda(6, x^3) = 1, \qquad \lambda^*(\mathbf{Z}_2, x^4) = \lambda(8, x^4) = 1,$$
$$\lambda^*(\mathbf{Z}_3, x^1) = \lambda(1, x^1) = 0, \qquad \lambda^*(\mathbf{Z}_3, x^2) = \lambda(3, x^2) = 0,$$
$$\lambda^*(\mathbf{Z}_3, x^3) = \lambda(6, x^3) = 1, \qquad \lambda^*(\mathbf{Z}_3, x^4) = \lambda(7, x^4) = 0,$$
$$\lambda^*(\mathbf{Z}_4, x^1) = \lambda(2, x^1) = 1, \qquad \lambda^*(\mathbf{Z}_4, x^2) = \lambda(4, x^2) = 1,$$
$$\lambda^*(\mathbf{Z}_4, x^3) = \lambda(6, x^3) = 1, \qquad \lambda^*(\mathbf{Z}_4, x^4) = \lambda(8, x^4) = 1.$$

3. *Asignarea elementelor intermediare.* Pentru asignarea elementelor intermediare cu respectarea condiției tehnice de bună funcționare se aplică algoritmul dat în 11.3. Sub formă tabelară automatul redus obținut este dat în tabelul 14.2.

Tabelul 14.2

Z \ x_1x_2	00	10	11	10
Z_1	Z_1 0	Z_1 0	Z_1 0	Z_3 0
Z_2	Z_2 1	Z_1 0	Z_4 1	Z_4 1
Z_3	Z_1 0	Z_3 0	Z_4 1	Z_3 0
Z_4	Z_2 1	Z_4 1	Z_4 1	Z_4 1

Mulțimea elementelor de trecere este $\Theta=\{\theta_1, \theta_2, \theta_3, \theta_4\}$, dată de mulțimea relațiilor de trecere

$$\mathscr{R}(\mathbf{A})=\{\theta_1=\mathbf{Z}_1\oplus Z_2,\ \theta_2=\mathbf{Z}_1\oplus\mathbf{Z}_3,\ \theta_3=\mathbf{Z}_2\oplus\mathbf{Z}_4,\ \theta_4=\mathbf{Z}_3\oplus\mathbf{Z}_4\}.$$

Impunînd condiția $\mathbf{Z}_1\oplus\mathbf{Z}_2\oplus\mathbf{Z}_3\oplus\mathbf{Z}_4=\theta_0$, se obține mulțimea relațiilor dintre elementele lui Θ, $\mathscr{R}(\Theta)=\{\theta_4=\theta_1, \theta_3=\theta_2\}$. Matricea de trecere rezultantă este dată în tabelul 14.3.

Luînd $\theta_1=01$, $\theta_2=10$ și $\mathbf{Z}_1=00$, se obține $\mathbf{Z}_2=01$, $\mathbf{Z}_3=10$, $\mathbf{Z}_4=11$, adică automatul cu funcția de tran-

ziție și funcția de ieșire din tabelul 14.4, unde prin u_1 și u_2 am notat variabilele asociate elementelor intermediare. Automatul obținut respectă condiția tehnică de bună funcționare.

Tabelul 14.3

Z \ x_1x_2	00	01	11	10
Z_1	θ_0	θ_0	θ_0	θ_2
Z_2	θ_0	θ_1	θ_2	θ_2
Z_3	θ_2	θ_0	θ_1	θ_0
Z_4	θ_2	θ_0	θ_0	θ_0

Tabelul 14.4

$\mathbf{Z}$ \ x_1x_2 / u_1u_2	00	01	11	10
$\mathbf{Z}_1$	00	00 0	00 0	00 0	10 0
$\mathbf{Z}_2$	01	01 1	00 0	11 1	11 1
$\mathbf{Z}_4$	11	01 1	11 1	11 1	11 1
$\mathbf{Z}_3$	10	00 0	10 0	11 1	10 0

4. *Determinarea funcțiilor de lucru ale elementelor intermediare și finale.* Fie U_1, U_2 elementele intermediare asignate, u_1, u_2 variabilele asociate acestora și $\mathscr{F}(U_1)$, $\mathscr{F}(U_2)$ funcțiile de lucru ale celor două elemente. Pentru determinarea expresiilor celor două funcții se va folosi algoritmul din 11.4. Ținînd seama de tabelul 14.4, se obțin următoarele expresii pentru funcțiile de lucru ale elementelor intermediare U_1 și U_2 :

$$\mathscr{F}(U_1) = x_1^0x_2^1u_1^1u_2^1 + x_1^1x_2^1u_1^0u_2^1 + x_1^1x_2^1u_1^1u_2^1 + x_1^1x_2^1u_1^1u_2^0 +$$
$$+ x_1^1x_2^0u_1^0u_2^0 + x_1^1x_2^0u_1^1u_2^0 + x_1^1x_2^0u_1^1u_2^1 + x_1^1x_2^0u_1^1u_2^0 + x_1^0x_2^1u_1^1u_2^0;$$
$$\mathscr{F}(U_2) = x_1^0x_2^0u_1^0u_2^1 + x_1^0x_2^0u_1^1u_2^1 + x_1^0x_2^1u_1^1u_2^1 + x_1^1x_2^1u_1^0u_2^1 +$$
$$+ x_1^1x_2^1u_1^1u_2^1 + x_1^1x_2^1u_1^1u_2^0 + x_1^1x_2^0u_1^0u_2^1 + x_1^1x_2^0u_1^1u_2^1.$$

În ceea ce privește elementul de ieșire Y, din analiza tabelului 14.4 se observă că are aceeași funcție de lucru ca și elementul U_2, $\mathscr{F}(Y)=\mathscr{F}(U_2)$.

5. *Simplificarea funcțiilor de lucru.* Pentru simplificare vom folosi unul dintre algoritmii dați în cap. 6. În general vom prefera metoda consensurilor, întrucît în această metodă nu sîntem obligați să plecăm de la expresia canonică a funcțiilor de lucru.

Să simplificăm mai întîi funcția $\mathscr{F}(U_1)$. Toate variabilele sînt biforme. Căutînd consensurile în raport cu fiecare dintre variabile, se obțin :

1°. Consensul în raport cu x_1 este $x_2^1 u_1^1 u_2^1$; după adăugarea consensului la $\mathscr{F}(U_1)$ și efectuarea absorbțiilor se obține noua expresia lui $\mathscr{F}(U_1)$:

$$\mathscr{F}(U_1) = x_2^1 u_1^1 u_2^1 + x_1^1 x_2^1 u_1^0 u_2^1 + x_1^1 x_2^1 u_1^1 u_2^0 + x_1^1 x_2^0 u_1^0 u_2^0 + \\ + x_1^1 x_2^0 u_1^1 u_2^0 + x_1^1 x_2^0 u_1^0 u_2^1 + x_1^1 x_2^0 u_1^1 u_2^1.$$

2°. Consensurile în raport cu x_2 sînt : $x_1^1 u_1^1 u_2^1$, $x_1^1 u_1^0 u_2^1$, $x_1^1 u_1^1 u_2^0$. Noua expresie a lui $\mathscr{F}(U_1)$ este

$$\mathscr{F}(U_1) = x_2^1 u_1^1 u_2^1 + x_1^1 u_1^1 u_2^1 + x_1^1 u_1^0 u_2^1 + x_1^1 u_1^1 u_2^0 + \\ + x_1^1 x_2^0 u_1^0 u_2^0 + x_2^1 u_1^1 u_2^0.$$

3°. Consensurile în raport cu u_1 sînt : $x_1^1 u_2^1$, $x_1^1 x_2^0 u_2^0$. Noua expresie a lui $\mathscr{F}(U_1)$ este

$$\mathscr{F}(U_1) = x_1^1 u_2^1 + x_1^1 x_2^0 u_2^0 + x_1^1 u_1^1 u_2^0 + x_2^1 u_1^1 u_2^1 + x_2^1 u_1^1 u_2^0.$$

4°. Consensurile în raport cu u_2 sînt : $x_1^1 x_2^0$, $x_1^1 u_1^1$, $x_2^1 u_1^1$. Noua expresie a lui $\mathscr{F}(U_1)$ este

$$\mathscr{F}(U_1) = x_1^1 u_2^1 + x_1^1 x_2^0 + x_1^1 u_1^1 + x_2^1 u_1^1.$$

Această expresie conține toți implicanții primi ai funcției $\mathscr{F}(U_1)$ și nu mai poate fi simplificată în continuare.

Procedînd în același mod, se obține următoarea expresie pentru $\mathscr{F}(U_2)$:

$$\mathscr{F}(U_2) = x_1^1 u_2^1 + x_2^0 u_2^1 + u_1^1 u_2^1 + x_1^1 x_2^1 u_1^1.$$

Funcția de lucru a elementului de ieșire Y este egală cu funcția de lucru a elementului U_2, $\mathscr{F}(Y) = \mathscr{F}(U_2)$, după cum se observă ușor din analiza tabelului 14.4.

În continuare ne vom ocupa mai întîi de sinteza dipolului corespunzător funcției de lucru a elementului U_1, $\mathscr{F}(U_1)$.

6. *Testarea realizării prin scheme fără repetiție și a condiției de ortogonalitate a codurilor asociate.* Pentru testarea realizării prin scheme fără repetiție se aplică algoritmul dat în 12.6. În primul rînd se aplică criteriul de ortogonalitate. Pentru aceasta este necesar să construim codurile grupale **K** și **K*** asociate funcției

$$\mathscr{F}_1(x_1, x_2, u_1, u_2) = x_1^1 u_2^1 + x_1^1 x_2^0 + x_1^1 u_1^1 + x_2^1 u_1^1$$

și dualei acesteia

$$\mathscr{F}_1^*(x_1, x_2, u_1, u_2) = x_1^1 u_1^1 + x_1^1 x_2^1 + x_2^0 u_1^1 u_2^1.$$

Codul asociat funcției $\mathscr{F}_1(x_1, x_2, u_1, u_2)$ este dat în tabelul 14.5, iar codul asociat funcției duale $\mathscr{F}_1^*(x_1, x_2, u_1, u_2)$ în tabelul 14.6. Ne referim la părțile din cele două tabele aferente coloanelor x_1^1, x_2^1, x_2^0, u_1^1, u_2^1 și p^1. Cu p^1 am notat latura sursei. Pentru codul **K** s-au înscris în tabelul 14.5 numai liniile pentru care p^1 are valoarea 1. În tabelul 14.6 este înscris întreg codul **K***. Trecînd la

Codul K — *Tabelul 14.5*

	x_1^1	x_2^1	x_2^0	u_1^1	u_2^1	p^1	a^1
A_1	1	0	0	0	1	1	0
A_2	1	0	1	0	0	1	0
A_3	1	0	0	1	0	1	1
A_4	0	1	0	1	0	1	0
A_5	1	0	1	1	1	1	1
A_6	0	1	1	1	1	1	0
A_7	0	1	0	0	1	1	1
A_8	0	1	1	0	0	1	1

*Codul K** — *Tabelul 14.6*

	x_1^1	x_2^1	x_2^0	u_1^1	u_2^1	p^1	a^1	w
B_2	1	0	0	1	0	1	1	1
B_1	1	1	0	0	0	1	0	2
B_3	0	0	1	1	1	1	0	4
B_4	0	1	1	0	1	1	1	
B_5	0	1	0	1	0	0	1	
B_6	1	0	1	0	1	0	1	
B_7	1	1	1	1	1	0	0	

verificarea relației de ortogonalitate între linii, se constată că aceasta nu este îndeplinită. Într-adevăr, este suficient să observăm că liniile A_3 și B_1 din codurile **K** și respectiv **K*** nu sînt ortogonale. Rezultă deci că funcția $\mathscr{F}_1(x_1, x_2, u_1, u_2)$ nu se realizează prin scheme fără repetiție.

7. *Determinarea funcției extinse.* Întrucît funcția nu se realizează prin scheme fără repetiție și în plus codurile

asociate nu sînt ortogonale, se trece la determinarea funcției de lucru extinse. În acest scop vom aplica algoritmul dat în 13.3. Se asociază primelor trei linii ale codului **K*** ponderile 1, 2 și respectiv 4. Aceste ponderi sînt trecute în coloana w din tabelul 14.6. Este suficient să luăm numai primele trei linii întrucît acestea formează o bază în **K***. Se calculează apoi rangurile variabilelor x_1^1, x_2^1, x_2^0, u_1^1, u_2^1 și p, folosind formula

$$m_i = \sum_{k=1}^{3} B_{ki} w_k,$$

m_i fiind rangul variabilei care corespunde coloanei i. De exemplu, m_4 este rangul variabilei u_1^1. Am folosit această notație pentru a simplifica scrierea.

Se obțin următoarele ranguri : $m_1 = 3$, $m_2 = 2$, $m_3 = 4$, $m_4 = 5$, $m_5 = 4$, $m_6 = 7$.

Vom verifica acum relația de ortogonalitate $\overset{6}{\underset{j=1}{\Xi}} A_{ij} m_j = 0$. Este suficient să testăm relația de ortogonalitate numai pentru primele patru linii din **K**, întrucît acestea formează o bază în **K**. Se vor obține :

$$\overset{6}{\underset{j=1}{\Xi}} A_{1j} m_j = 0, \quad \overset{6}{\underset{j=1}{\Xi}} A_{2j} m_j = 0, \quad \overset{6}{\underset{j=1}{\Xi}} A_{3j} m_j = 1, \quad \overset{6}{\underset{j=1}{\Xi}} A_{4j} m_j = 0.$$

Nu se verifică una dintre relațiile de ortogonalitate. Pentru înlăturarea acestui inconvenient vom introduce o nouă variabilă de rang 1, egal cu valoarea relației de neortogonalitate pe care o notăm prin a. Variabilei a îi corespunde ultima coloană în tabelele 14.5 și 14.6. Determinarea elementelor coloanei a din tabelul 14.5 se face astfel : în liniile pentru care relația de neortogonalitate are valoarea egală cu rangul lui a se trece valoarea 1, iar în restul de linii se trece valoarea 0. Elementele din liniile $A_5 - A_8$ ale coloanei a se obțin prin compunerea elementelor din primele patru linii, la fel cum s-au obținut liniile $A_5 - A_8$ din liniile $A_1 - A_4$, pentru restul de coloane.

Coloana a din tabelul 14.6 se determină prin dualitate din tabelul 14.5, întrucît cele două tabele conțin codurile **K*** și **K**. Pentru aceasta este suficient să testăm ortogonalitatea liniei A_3 din **K** cu liniile codului **K*** și să luăm valorile convenabile pentru elementele coloanei a astfel încît să se verifice ortogonalitatea. Se obține rezultatul din tabelul 14.6.

Expresiile extinse ale funcțiilor $\mathscr{F}(U_1)$ și $\mathscr{F}^*(U_1)$, duala lui $\mathscr{F}(U_1)$, după introducerea variabilei a devin :

$$\Phi_1(x_1, x_2, u_1, u_2, a) = a^1 u_2^1 + x_1^1 x_2^0 + x_1^1 u_1^1 a^1 + \\ + x_2^1 u_1^1 + x_2^1 u_2^1 a^1 + x_2^1 x_2^0 a^1,$$

$$\Phi_1^*(x_1, x_2, u_1, u_2, a) = x_1^1 u_1^1 a^1 + x_1^1 x_2^1 + x_2^0 u_1^1 u_2^1 + x_2^1 x_2^0 u_2^1 a^1.$$

În expresiile lui $\Phi_1(x_1, x_2, u_1, u_2, a)$ și $\Phi_1^*(x_1, x_2, u_1, u_2, a)$ am reținut și implicanții nuli $x_2^1 x_2^0 a^1$ și respectiv $x_2^1 x_2^0 u_2^1 a^1$, întrucît acest lucru are importanță pentru sinteza dipolului corespunzător, variabilele x_2^1 și x_2^0 fiind considerate independente din punctul de vedere al sintezei.

Am ajuns astfel la o expresie a funcției $\mathscr{F}(U_1)$ căreia îi corespunde un cod grupal.

8. *Întoarcere la testarea realizării prin scheme fără repetiție.* Este necesar să ne întoarcem la pasul 6 al algoritmului, testarea realizării prin schme fără repetiție. În acest caz nu mai este suficient testul de ortogonalitate a liniilor din codurile **K** și **K***, întrucît acest test se verifică. Este necesar să aplicăm complet algoritmul din 12.6.

Mai întîi să observăm că variabilelor x_2^0 și u_2^1 le corespund coloane identice în **K***. Rezultă că x_2^0 și u_2^1 sînt conectate în paralel. Conform observației de la teorema 2 din 12.6 este necesar să înlăturăm una dintre aceste coloane pentru aplicarea algoritmului. Să presupunem că înlăturăm coloana corespunzătoare lui u_2^1. Construim codul $\mathbf{K}_2^*$ și reținem numai liniile din $\mathbf{K}_2^*$ cu proprietatea că au un singur element egal cu 1, restul fiind nule. Aceste linii sînt următoarele : $C_1 = B_2 B_3$, $C_2 = B_2 B_5$, $C_3 = B_2 B_6$, $C_4 = B_3 B_5$, $C_5 = B_3 B_6$ și $C_6 = B_5 B_6$. C_1 are 1 în coloana 6 și corespunde lui p^1, C_2 corespunde lui x_2^1, C_3 lui x_1^1, C_4 lui u_1^1, C_5 lui x_2^0 (deci și lui u_2^1) și C_6 lui a^1.

9. *Sinteza dipolului fără repetiție.* Ne găsim în condițiile teoremei 4 din 12.6. Există șase linii $C \in \mathbf{K}_2^*$, adică atîtea cîte laturi neconectate în paralel există și patru linii $B \in \mathbf{K}^*$, astfel încît fiecare dintre cele patru linii B apare în două și numai două produse C. Se poate trece deci la sinteza dipolului fără repetiție, adică la pasul 7 al algoritmului.

Există o singură schemă fără repetiție care realizează funcția dată. Această schemă, reprezentată în fig. 14.3,

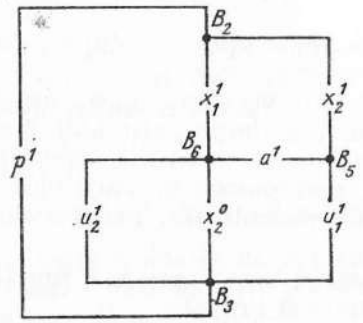

Fig. 14.3

are patru noduri B_2, B_3, B_5 și B_6 și șapte laturi x_1^1, x_2^1, x_2^0, u_1^1, u_2^1, p și a, din care două x_2^0 și u_2^1 sînt conectate în paralel.

10. *Determinarea variabilelor care se repetă.* Întrucît funcția $\mathscr{F}(U_1)$ a fost extinsă, este necesar să stabilim care dintre variabile se repetă, aplicînd pasul 10 al algoritmului. În acest scop rezolvăm ecuația

$$\Phi_1(x_1, x_2, u_1, u_2, a) = \mathscr{F}_1(x_1, x_2, u_1, u_2),$$

unde am făcut notația

$$\mathscr{F}_1(x_1, x_2, u_1, u_2) = \mathscr{F}(U_1),$$

echivalentă cu ecuația

$$\Phi_1(x_1, x_2, u_1, u_2, a)\,\overline{\mathscr{F}}_1(x_1, x_2, u_1, u_2) + \overline{\Phi}_1(x_1, x_2, u_1, u_2, a)\,\mathscr{F}_1(x_1, x_2, u_1, u_2) = 0.$$

Întrucît $\overline{\mathscr{F}}_1(x_1, x_2, u_1, u_2) = x_1^0 x_2^0 + x_1^0 u_1^0 + x_2^1 u_1^0 u_2^0$ și

$$\overline{\Phi}_1(x_1, x_2, u_1, u_2, a) = x_1^0 x_2^0 + x_1^0 u_1^0 u_2^0 + x_1^0 u_1^0\, a^0 + x_2^1 u_1^0 u_2^0,$$

ecuația de mai sus devine

$$x_1^0 u_1^0 x_2^1 u_2^1 a^1 = 0$$

care admite soluția $a = (0,\ x_1^1 + u_1^1 + x_2^0 + u_2^0)$. Rezultă că se pot obține cinci soluții simple pentru a: $a = 0$, $a = x_1^1$, $a = u_1^1$, $a = x_2^0$, $a = u_2^0$. În funcție de tipul elementelor de care dispunem se poate alege una dintre soluții.

Este interesant că apare și o soluție în care a poate să lipsească, $a = 0$. Acest caz se întîmplă din cauză că implicantul $x_1^1 u_1^1$ este consensul implicanților $x_1^1 x_2^0$, $x_2^1 u_1^1$.

Cu aceasta sinteza multipolului este terminată. Dipolul corespunzător funcției de lucru a elementului U_1 se obține luînd ca poli nodurile de conectare a laturii p și înlăturînd latura p.

Trecem acum la sinteza dipolului corespunzător funcției de lucru a elementului U_2.

11. *Testarea realizării prin scheme fără repetiție și a condiției de ortogonalitate pentru funcția* $\mathscr{F}(U_2)$. Se aplică pasul 6 al algoritmului la funcția

$$\mathscr{F}(U_2) = \mathscr{F}_2(x_1, x_2, u_1, u_2) = x_1^1 u_2^1 + x_2^0 u_2^1 + u_1^1 u_2^1 + x_1^1 x_2^1 u_1^1.$$

Codurile asociate funcției $\mathscr{F}_2(x_1, x_2, u_1, u_2)$ și dualei acesteia $\mathscr{F}_2^*(x_1, x_2, u_1, u_2) = x_1^1 u_2^1 + x_2^1 u_2^1 + u_1^1 u_2^1 + x_1^1 x_2^0 u_1^1$ sînt date în tabelele 14.7 și respectiv 14.8. Am înscris numai liniile corespunzătoare implicanților primi ai celor două funcții.

Tabelul 14.7

	x_1^1	x_2^1	x_2^0	u_1^1	u_2^1	p^1	a^1	b^1	c^1	m
A_1	1	0	0	0	1	1	0	0	1	8
A_2	0	0	1	0	1	1	0	0	0	0
A_3	0	0	0	1	1	1	0	1	0	2
A_4	1	1	0	1	0	1	1	0	0	1

Asociind liniilor din tabelul 14.8 ponderile din coloana w, se obțin următoarele ranguri pentru cele șase variabile $m_1 = 9$, $m_2 = 1$, $m_3 = 4$, $m_4 = 3$, $m_5 = 14$, $m_6 = 15$.

Tabelul 14.8

	x_1^1	x_2^1	x_2^0	u_1^1	u_2^1	v^1	w
B_1	1	0	0	0	1	1	8
B_2	0	1	0	0	1	1	4
B_3	0	0	0	1	1	1	2
B_4	1	0	1	1	0	1	1

Verificînd relațiile de ortogonalitate, se obține :

$$\sum_{j=1}^{6} A_{1j} m_j = 9 \oplus 14 \oplus 15 = 8,$$

$$\sum_{j=1}^{6} A_{2j} m_j = 1 \oplus 14 \oplus 15 = 0,$$

$$\sum_{j=1}^{6} A_{3j} m_j = 3 \oplus 14 \oplus 15 = 2,$$

$$\sum_{j=1}^{6} A_{4j} m_j = 9 \oplus 4 \oplus 3 \oplus 15 = 1.$$

Este necesar deci să introducem trei variabile suplimentare a^1, b^1, c^1 cu ponderile egale respectiv cu 1, 2 și 8. Elementele corespunzătoare celor trei variabile sînt trecute în ultimele trei coloane ale tabelului 14.7.

Se trece la testarea independenței liniare între coloanele noi introduse și coloanele vechi din codul **K** (tabelul 14.7). Se constată că a^1 este identic cu x_2^1, deci sînt contectate în serie și se poate renunța la el. Există și o relație de dependență liniară între coloanele x_1^1, u_1^1, b^1 și c^1, $x_1^1 \oplus u_1^1 \oplus b^1 \oplus c^1 = \theta_0$, unde θ_0 este coloana cu toate elementele nule.

Se poate renunța la o variabilă suplimentară. Acest lucru se poate realiza prin introducerea variabilei $d^1 = x_1^1 \oplus c^1 = u_1^1 \oplus b^1$ și prin înlocuirea produselor $x_1^1 c^1$ și $u_1^1 b^1$ prin d^1 în implicanții primi $x_1^1 u_2^1 c_1$ și respectiv $u_1^1 u_2^1 b^1$. Se obține următoarea expresie pentru funcția extinsă :

$$\Phi_2(x_1, x_2, u_1, u_2, d) = d^1 u_2^1 + x_2^0 u_2^1 + x_1^1 x_2^1 u_1^1.$$

Se va verifica apoi dacă $\Phi(x_1^1, x_2^1, x_2^0, u_1^1, u_2^1, d)$ se realizează prin scheme fără repetiție. Se va constata că acest lucru se întîmplă, multipolul rezultat fiind reprezentat în fig. 14.4. Nu vom mai face această verificare aici, întrucît se repetă algoritmul folosit anterior.

12. *Determinarea variabilelor care se repetă.* În acest scop rezolvăm ecuația

$$\mathcal{F}_2(x_1, x_2, u_1, u_2) = \Phi_2(x_1, x_2, u_1, u_2, d),$$

adică

$$x_1^1 u_2^1 + x_2^0 u_2^1 + u_1^1 u_2^1 + x_1^1 x_2^1 u_1^1 = d^1 u_2^1 + x_2^0 u_2^1 + x_1^1 x_2^1 u_1^1.$$

După simplificări se obține

$$u_2^1 x_2^1 [d^0 (x_1^1 u_1^0 + x_1^0 u_1^1) + d^1 x_1^0 u_1^0] = 0$$

și pentru $u_2^1 = x_2^1 = 1$, ecuația devine

$$d^0 (x_1^1 u_1^0 + x_1^0 u_1^1) + d^1 x_1^0 u_1^0 = 0,$$

care are soluția

$$d = (x_1^1 u_1^0 + x_1^0 u_1^1, x_1^1 + u_1^1).$$

Vom lua $d = x_1^1 + u_1^1$ care conduce la schema cea mai simplă în care se repetă variabilele x_1^1 și u_1^1. Nu vom mai

reprezenta această schemă întrucît ea se obține ușor din schema din fig. 14.4 prin înlocuirea lui d cu variabilele x_1^1 și u_1^1, conectate în paralel.

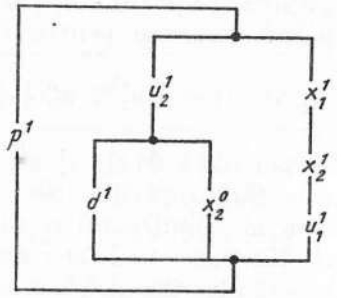

Fig. 14.4

Schema automatului este dată în fig. 14.5, unde a poate lua una oricare dintre valorile 0, x_1^1, u_1^1, x_2^0 sau u_2^0.

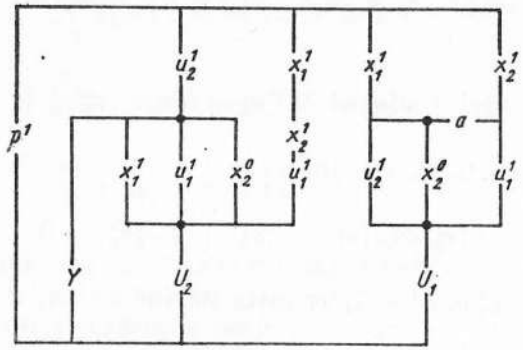

Fig. 14.5

1. Benzaken, C. *Contributions des structures alg{briques ordonnées à la theorie des reseaux*. These, Faculte des sciences de l'Universitc de Grenoble, (1968).
2. Chein, M. *Etude des decompositions d'un reseau. Application à l'ecriture des fonctions booleennes en sommes et produits*. Grenoble, 1967.
3. Debraine, P. *Machines de traitement de l'information (circuits et programmes)*. Tome I, *Etude logique et construction des circuits*. Paris, Sofradel Masson et Cie, 1967.
4. Demidovici, H. B. *K teorii gruppovih kodov*. În : Problemi kibernetiki, 1961.
5. Gluşkov, V. M. *Abstraktnaia teoria avtomatov*. În : Uspehi matematiceskih nauk, **16**, 5, 1961.
6. Gluşkov, V. M. *Sintez ţifrovih avtomatov*. Moscova, Fizmatghiz, 1962.
7. Ioanin, Gh. *Contributions à l'étude du fonctionnement des circuits a contacts réels*. București, *Simpozionul Hazard and race phenomena in switching circuits*, 1968.
8. Ioanin, Gh. *Asupra teoriei algebrice a contactelor multipoziționale și aplicațiile ei la studiul contactelor reale*. În : B-ul Științ. Acad. R. P. R., Secția mat. fiz., **7**, 2, 1955, p. 231.
9. Ioanin, Gh. *Bestimmung der minimalen Anzahl der zustande für einen Asynchron-automaten*. În : Revue Roumaine de mathématiques pures et appliquées, **14**, 4, 1969.
10. Kuntzmann, J. *Théorie des relations et des reseaux*, Grenoble, Institut de Mathématiques appliquées, 1966.
11. Lapscher, F. *Fermetures sur l'ensemble des fonctions booléennes incompletes. Applications*, Grenoble, Institut de Mathématiques Appliquées, 1968.
12. Livovschi, L. *Automate finite cu elemente logice pneumatice și hidraulice*. București, Editura, Academiei, 1963.
13. Livovschi, L. *Circuite cu contacte cu relee*. București, Editura Academiei, 1968.
14. Livovschi, L. *O problemă de sinteză a automatelor finite*. În : Studii și cercetări matematice, **18**, 8, 1966.
15. Livovschi, L. *Minimizarea automatelor asincrone*. În : Studii și cercetări matematice, **17**, 8, 1965.
16. Moisil, Gr. C. *Teoria algebrică a mecanismelor automate*. București, Editura tehnică, 1959.
17. Moisil, Gr. C. *Scheme cu comandă directă cu contacte și relee*. București, Editura Academiei, 1959.
18. Moisil, Gr. C. *Funcționarea în mai mulți timpi a schemelor cu relee ideale*. București, Editura Academiei, 1960.
19. Moisil, Gr. C. *Circuite cu tranzistori*, I. București, Editura Academiei, 1961.

20. Moisil, Gr. C. *Circuite cu tranzistori*, II. București, Editura Academiei, 1962.
21. Moisil, Gr. C. *Funcționarea reală a schemelor cu contacte și relee*. București, Editura Academiei, 1965.
22. Moisil Gr. C. *Théorie structurelle des automates finis*. Paris, Gauthier-Villars, 1967.
23. Moisil, Gr. C. *Sur les aleas de continuité dans les circuits de comutations*. În : Automatisme, **10**, 9, 1965.
24. Moisil, Gr. C. *Logique, Automatique, Informatique. Les états transitoires dans les circuits séquentiels*, București, Editura Academiei, 1972.
25. Muntean, I. *Asupra problemei sintezei schemelor cu contacte fără repetiție*. În : Probleme de automatizare, 5, 1962, p. 115—125.
26. Muntean, I. *Metod rasciota elektriceskih țepei*. În : Revue de mathématiques pures et appliquées, **8**, 1, 1963, p. 117—137.
27. Muntean, I. *Contribuții la rezolvarea problemei sintezei schemelor cu contacte și relee* (I). În : Studii și cercetări matematice, **12**, 1, 1964, p. 477—497.
28. Muntean, I. *Contribuții la rezolvarea problemei sintezei schemelor cu contacte și relee* (II). În : Studii și cercetări matematice, **18**, 1, 1966, p. 87—131.
29. Muntean I. *Un criteriu de recunoaștere a funcțiilor booleene care se realizează prin scheme fără repetiție*. București, Sesiunea generală științifică a Academiei Republicii Socialiste România, 1966.
30. Muntean, I. *Asupra teoriei codurilor grupale și aplicațiile lor la sinteza multipolilor cu contacte și relee*. În : Studii și cercetări matematice, **18**, 9, 1966, p. 1247—1257.
31. Muntean, I. *Simplificarea funcțiilor de variabile polivalente*. În : Studii și cercetări matematice : **19**, 2, 1967, p. 241—257.
32. Muntean, I. *Sinteza automatelor finite cu număr minim de stări*. În : Studii și cercetări matematice, **20**, 1. 1968, p. 3—13.
33. Muntean, I. *Sistem de algoritme pentru sinteza automatelor finite*. Brașov, Simpozionul de teoria sistemelor, 1973.
34. Muntean, I. *Algoritm de asignare automată a elementelor intermediare în automatele finite*. În : Probleme de automatizare, 8, 1974, p. 287—294.
35. Naslin, P. *Circuite logice și automatizări secvențiale* (trad. din l. franceză). București, Editura tehnică, 1967.
36. Necula, N. *Circuite logice : sinteza automată. Circuite multiprag*. București, Editura Academiei, 1972.
37. Petrick, S. R. *A direct decomposition of the irredundant forms of a Boolean function from the set of prime — implicants* În : Techn. reports Air Force Cambridge Research Center, (1956), p. 56—110.
38. Pichat, E. *Decompositions et ecriture minimales des fonctions booléennes*. București, Colocviul *Tehnici de calcul și calculatoare*, 1967.
39. Prudhomme, R. *Automatique*, I. *Systèmes séquentiels à niveau, Systèmes asservis linéaires continus*. Paris, Masson et Cie, 1970.
40. Quine, W. V. *The problem of simplifying of truth functions*. În : Amer. Math. Monthly, **59**, 8, 1952, 521—531.

41. Rudeanu, S. *Axiomele laticelor și algebrelor booleene*. București, Editura Academiei, 1963.
42. Rudeanu, S. *Asupra rezolvării ecuațiilor booleene prin metoda lui Löwenheim*. În : Studii și Cercetări matematice, **13**, 2, 1962.
43. Saucier, G. *Algorithme practique de codage des automates asynchrones*. Grenoble, Institut de Mathematiques Appliquées, 1968.
41. Tison, P. *Théorie des consensus,, Recherche des bases premières d'une fonction booléenne*. În : Automatisme, 10, iunie, 1965, p. 229—234.
45. Vallée, R. L. *Analyse binaire*, I. *Théorie et applications aux circuits combinatoires*, Paris, Masson et Cie, 1970.
46. Vallée, R. L. *Aanlyse binaire*, II. *Clef des automates numeriques* Paris, Masson et Cie, 1970.